Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

An integrative view of foveated rendering

Bipul Mohanto[a,*], ABM Tariqul Islam[a,*], Enrico Gobbetti[b,**], Oliver Staadt[a,*]

[a] *Institute for Visual & Analytic Computing, University of Rostock, Germany*
[b] *Visual and Data-intensive Computing, CRS4, Italy*

ARTICLE INFO

Article history:
Received June 1, 2023

Keywords:
Foveated rendering, Gaze-contingent rendering, Adaptive resolution, Geometric simplification, Shading simplification, Chromatic degradation, Spatio-temporal deterioration

ABSTRACT

Foveated rendering adapts the image synthesis process to the user's gaze. By exploiting the human visual system's limitations, in particular in terms of reduced acuity in peripheral vision, it strives to deliver high-quality visual experiences at very reduced computational, storage, and transmission costs. Despite the very substantial progress made in the past decades, the solution landscape is still fragmented, and several research problems remain open. In this work, we present an up-to-date integrative view of the domain from the point of view of the rendering methods employed, discussing general characteristics, commonalities, differences, advantages, and limitations. We cover, in particular, techniques based on adaptive resolution, geometric simplification, shading simplification, chromatic degradation, as well spatio-temporal deterioration. Next, we review the main areas where foveated rendering is already in use today. We finally point out relevant research issues and analyze research trends.

1. Introduction

Over the past decade, both the display resolution and pixel density have rapidly increased in response to the demands of a variety of application setups, including immersive virtual reality (VR), augmented reality (AR), mixed reality (MR), and large high-resolution displays (LHRD). Despite the impressive improvements witnessed in the past, current displays are still far from matching human capabilities, and growth in pixel counts and density is still continuing. For instance, the densest commercial near-eye displays can offer an angular resolution on an average of 10-15 cycles per degree, with exceptions such as Varjo VR-3 achieving angular resolution of 35 cycles per degree [213, 45], while humans can perceive over 60 cycles per degree in the fovea centralis [200, 169]. Moreover, current displays are also viewing-angle restricted, e.g., on average VR displays are limited to a field-of-view (FOV) of $90^\circ - 110^\circ$ [45] whereas a human can perceive a much wider range (see Sec. 3.2). Moreover, while some commercial displays have appeared that significantly increase FOVs (e.g., StarVR reaches a 210° horizontal FOV), supporting wide FOVs together with high resolution is an open research problem[251]. Specific setups, like stereoscopic or light field displays, further increase the needed pixel count.

Interactive and immersive applications must also meet the important constraints on refresh rates imposed by the human perceptual system. Nowadays, 90 Hz has been established as a standard VR frame rate, while interactive gaming monitors maintain $\geq$ 120 Hz [88]. Nevertheless, according to Cuervo et al. [50], the refresh rate may need to be increased up to 1800 Hz for life-like VR immersion.

The need to generate a large number of pixels at very high frequencies is only partially matched by the concurrent increase in the performance of graphics hardware. First of all, the hardware capabilities are typically ex-

*Corresponding authors: bipul.mohanto@uni-rostock.de, a.islam@uni-rostock.de, oliver.staadt@uni-rostock.de
**Corresponding author: enrico.gobbetti@crs4.it

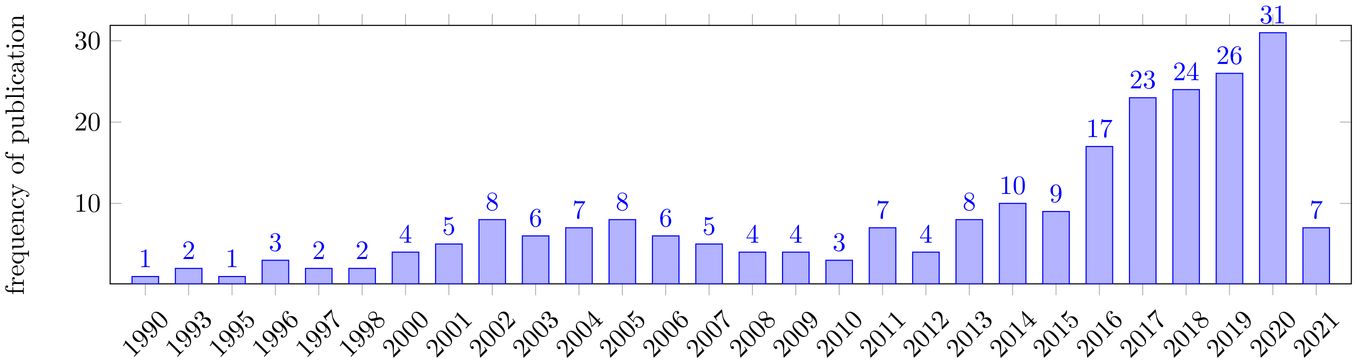

Fig. 1. The chart is depicting the outstanding foveated rendering research included in this survey report. The novel techniques described different peripheral degradation techniques. The latest papers are still unpublished in 2021 during writing this survey.

ploited to improve the visual realism of rendered images, by increasing scene complexity or rendering quality. Many data sets, including large simulation data [104], CAD models [257], or production-quality 3D scene descriptions [212] are often exceedingly large and costly to render in even the simplest modality. Moreover, while global illumination algorithms, such as ray tracing and path tracing have been significantly accelerated in the recent years by the emergence of programmable GPUs with general-purpose programming capabilities and dedicated raytracing cores, real-time photorealistic image synthesis remains extremely difficult on current graphics platforms because of the intrinsic complexity of accurately computing light propagation in complex and possibly dynamic environments. Scaling to remote rendering systems is only a very partial solution since video transmission matching human visual field and frequency constraints consumes over 100 Gbps [22], which is infeasible over the current network standard.

As a result, generating high-quality interactive experiences remains an elusive target that we cannot expect to solve in the foreseeable future by hardware performance improvement alone. For this reason, the last decades have seen a flourishing of methods that strive to improve rendering performance in time and resource-constrained settings [257, 6]. The underlying idea of all these techniques is to exploit various characteristics of our visual system to present approximate images that can be computed or transmitted with the available resources and timing constraints while being perceived identical, or marginally different, to the high-quality target.

In particular, on displays that uniformly cover a reasonably large FOV, much of the visual information is wasted due to the space-variant nature of human vision, which has high resolution only in a small central region. In fact, due to the highest cone density, the color and visual detail perception are higher in a smaller retinal region, the fovea [106, 38, 80]. Aside from the fovea, vision in the periphery quickly diminishes. As a result, in current VR setups, only 4% of the pixels are visible at a fixation [106, 173]. Likewise, Wei et al. [238] report foveated region covers roughly 8% of the whole 60° of a desktop monitor.

Developing specialized image synthesis methods that exploit the human visual system's limitations, in particular in terms of reduced acuity in peripheral vision, to deliver high-quality visual experiences at very reduced computational, storage, and transmission costs is thus a potentially very effective approach. Techniques to achieve this goal have been introduced in the past under the name of "foveated rendering" [80, 173], "gaze-contingent rendering" [61, 56, 34, 60, 163, 151, 218, 203, 204, 9, 26, 239, 114] or, in more general context, "perception driven rendering" [164, 27]. However, "foveated rendering" is more prevalent in the literature. Thereby, in this survey, we will stick to this terminology. Over the years, many foveated rendering techniques have been introduced to optimize rendering fidelity, frame rate, compression, transmission, and power consumption (Fig. 1). In this context, the fundamental tasks are the identification of the user's gaze and the exploitation of this knowledge to perform the optimization. Many variations have been proposed, with vertical solutions dependent on specific gaze tracking, displays, or rendering algorithms.

In the recent past, several surveys have been presented in foveated rendering research (Sec. 2). However, these studies were mainly limited to particular display technologies (mostly VR), applications, as well as on perceptual issues. On the other hand, our survey provides an up-to-date integrative view of foveated rendering, investigating the entire research spectrum from the point of view of the *rendering methods employed*, showing their commonalities, differences, and specialization to specific setups. Compression and transmission are covered as they form an enabling technology for distributed rendering. The target audience of our survey includes computer graphics researchers and practitioners in relevant application fields. Researchers will find a structured overview of the field, which organizes the various problems and existing solutions, classifies the existing literature, and indicates challenging open problems. Practitioners and domain experts will, in turn, find a presentation of the areas where foveated rendering has already been applied in practice, as well as an analysis of

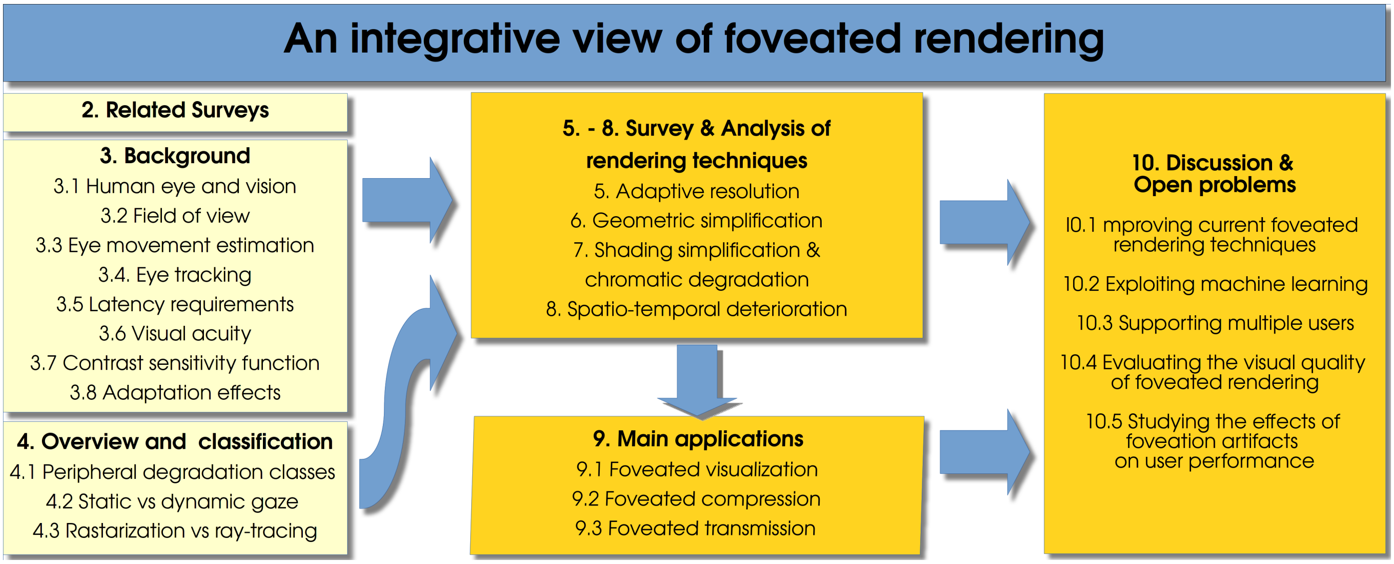

Fig. 2. A visual index of this survey.

applications and settings that still pose major challenges.

After summarizing the related survey literature (Sec. 2), we present an overview of relevant properties of the human visual system (HVS) and explain the different terminologies required to comprehend the foveated rendering (Sec. 3). Following that, we provide an abstract characterization of the techniques that can be applied for foveated rendering, introducing our proposed classification (Sec. 4). The various solutions proposed in the literature, their fundamental elements, key problems, as well as promising potential research directions are then analyzed according to our classification (Sec. 5 - 8). We then provide an overview of the main applications in which foveated rendering has been applied (Sec. 9). We finally discuss the identified research issues and research trends (Sec. 10) and conclude with a general summary of the findings of this study (Sec. 11). A visual index of this survey is depicted in Fig. 2.

2. Related surveys

The study of foveation effects has a very long history. Early applications were mostly in psychophysical research, with experiments centered around studying the effects of stimuli presented when the participant's gaze is fixated upon a predefined location. Such a concept was first proposed by Aubert and Foerster in 1857 [15]. Later, in 1973, Stephen Reader [185] was among the first to develop computerized *gaze-contingent imagery*. Following, the gathered knowledge was exploited in a variety of applications, giving birth to the foveated rendering research area. Extensive surveys on different facets of foveated rendering have been conducted over time, such as eye-tracking [180, 190, 107, 191, 33, 119], latency requirements [8, 218, 135, 221], foveated display classification [102, 202, 61, 171], gaze-contingent rendering [229], peripheral vision [206], peripheral limitations [83], peripheral degradation effect [235], peripheral visual artifacts [97], graphics quality constraints [40], foveated path tracing [118], foveated VR and AR optics [45, 95]. However, an up-to-date overall characterization and study of the graphics techniques employed for optimization purposes are missing.

In an eye-tracking and interaction survey, Duchowski et al. [215] propose a taxonomy for gaze-based interaction applications in which foveated rendering has been described as a *passive interaction* that manipulates the screen content in response to eye movement. The taxonomy further is classified into *model* and *image-based* rendering. The model-based approaches pre-manipulate graphics geometry before even the rendering process starts, e.g., number of triangles reduction. In contrast, the image-based approaches reduce spatiotemporal complexity of pixel data just before rendering with convolution filter, e.g., Laplace [34], Gaussian [223, 42, 140], and Kalman filter [96]. Noteworthy, the Gaussian filter is widely used as it is more compatible with the human visual system [42]. This taxonomy has been well adopted in several other studies [57, 58]. Furthermore, Hunter et al. [163] combine both image and model-based rendering as a *hybrid approach* which is more appropriate for GPU implementation on modern hardware. In another survey on gaze-contingent display, Duchowski et al. [61] classify screen-based foveated rendering into *focus plus context* and *screen-based* displays. Spjuit et al. [102, 202] provide a classification of displays along two axes. The first one characterizes a display according to how angular resolution varies as a function of eccentricity. The second axis, addresses how a system adapts to changes in user gaze direction. As each of these axes is divided in four categories (from none to full), a total of 16 display categories are identified.

Among the most relevant surveys, Swafford et al. [210] investigate four foveated rendering methods: peripheral resolution, variable per-pixel depth buffer samples for screen-space ambient occlusion (SSAO), GPU-level tessellation for the fovea, and variable-per-pixel ray casting measures throughout the field of view. Weier et al. [244] concisely surveyed foveated rendering in the context of the more general field of "perception-driven rendering". In this survey, foveated rendering has been classified into two classes: with and without an active gaze tracker, and further divided into scene simplification and adaptive sampling. The work has been further extended in Martin Weier's Ph.D. thesis [239], which extends the previous state-of-the-art report [244] discussing pre-filtering, sampling adaptation, temporal coherence, and post-filtering aspects of current perception-driven methods. Our work focuses exclusively on foveated graphics and provides a deeper coverage of this field. Most recently, Matthews et al. [150] published a brief report on a few seminal foveated rendering research, with existing research challenges and future research directions. Noteworthy, most of these surveys are strictly limited to VR displays.

In contrast to previous studies, our survey does not target a particular display technology or application. This review aims to investigate the entire foveated rendering research spectrum, focusing on characterizing the classes of optimization methods employed and showing their specializations to different settings, from near-eye displays to large high-resolution displays and application domains.

3. Background

Foveated rendering, similarly to other approximate rendering techniques, aims to optimize various aspects of the rendering process by exploiting the peculiar characteristics of the human visual system. In this section, we provide relevant background information to create a common ground for concepts and conventions used in the rest of the paper.

3.1. Human eye and vision

The human visual system (HVS) is a complex biological system that contains 70% of all photoreceptors and four billion neurons. Almost half of the primary visual cortex is engaged in vision [48] in which 25% is devoted to processing data from central visual angle (2.5°) [101]. The eye works as a vision sensor that allows light rays to pass to the retina through an adjustable iris, being refracted by the cornea and a crystalline lens using six different muscle movements [45, 244]. The retina consists of three types of photoreceptors: rods, cones, and retinal ganglion cells, which convert the light signal into an electrical signal. The optical nerves work as *information bus* which transmit visual signals from the retina to the visual cortex with an estimated bandwidth of 10 Mbps [22]. The rods are highly light-sensitive and even can be activated by a single photon. Cones are, on the other hand, less light-sensitive but pass color and detailed visual cues to the visual cortex for further processing.

There are approximately 120 million rods, six million cones, and 24-60 thousand photosensitive retinal ganglion cells [43]. Noteworthy, these numbers may vary in different studies, e.g., Kaplanyan et al. [106] suggest 4.6 million cones. The cones have a high density around the center of the optical axis known as the fovea; around 1.50 mm in diameter [198]. Different studies have revealed distinct foveal angles in between $2° - 5.2°$ around the optical axis [80, 156, 173]. The HVS processes the highest acuity of contrast, color, and depth information in the fovea [26]. The neighboring regions in a circle of up to 8° is called parafovea, and up to 17° perifovea. Exceeding that begins the peripheral region [190, 121], which can be further classified as near, mid, and far peripheral (see Fig. 3) [16]. Most foveated rendering solutions differentiate among the central foveal region, where most of the rendering effort is concentrated, and the rest of the field-of-views [80, 156, 173].

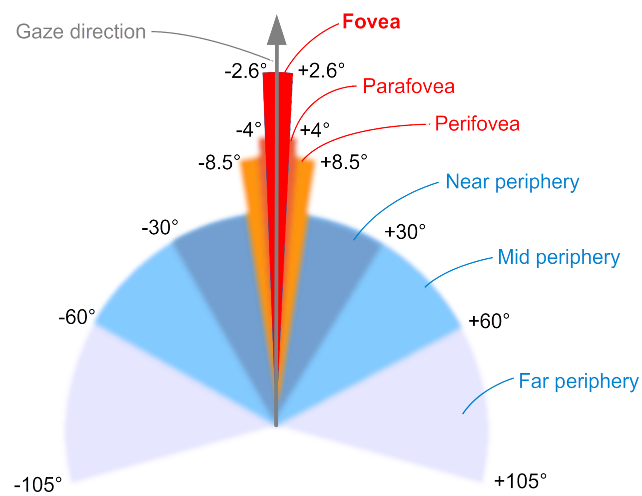

Fig. 3. The foveal angle varies between studies. However, most studies mention from $\pm 2°$ to $\pm 5.2°$ around the optical axis [80, 156, 173]. Neighboring regions in a circle of size of 8° is called parafovea, and up to 17° perifovea. Exceeding that begins the peripheral region [190, 121], which can be further classified into near (until $\pm 30°$ around the optical axis), mid (from $\pm 30°$ to $\pm 60°$), and far (from $\pm 60°$ to $\pm 105°$) periphery [16]. Human vision roughly spans $\pm 105°$ horizontally around the gaze direction when the head is stable [11]

.

3.2. Field of view

The human vision spans roughly $210° \times 135°$ [11]; however, this measure with a steady focus of the eyes. The stereoscopic vision is composed of two monocular visions, which the brain stitches together. Each eye has roughly $162° - 165°$ monoscopic field of view (FOV), and $\approx 114°$ overlap region [80]. Nonetheless, with head rotation, humans can see almost $270° - 290°$ horizontal arc. However, physically, humans can roughly observe only around 90° in as little as 1/10 second during saccades and can follow moving objects at speed up to $180°/s$ [50]. Under a near-eye VR display, the immersion consistently begins from

($\approx 80°$) FOV and steadily grows up with higher angle [118] whereas higher eccentricity raises the risk of motion sickness. Furthermore, there is an existing research challenge between FOV and angular resolution. The increment of FOV lowers the angular resolution which may easily perceivable by the viewers.

3.3. Eye movement estimation

The eye movements, such as *saccades, smooth pursuit, vergence and accommodation*, and *vestibulo-ocular movements* directly affect human perception [142]. The saccades are the rapid ballistic eye motions that suddenly disrupt intervals of fixation and lead the fovea to the scene's *region of interest (ROI)*, and lasts 10-100 ms exceeding $300°/s$ [53, 181, 142, 150]. Perceptual changes during brief saccades are barely detectable by humans [187]. Smooth pursuit is active during eyes track a moving object with detectable velocity. Vergence and accommodation refer to the eye's fixation process, in which the ciliary muscles change the crystalline lens's refractive potential to reduce the volume of a blur for the fixated depth of the scene [89]. Vestibulo-ocular movement occurs while the eye is locked on an ROI, but the head moves. For more details about eye movements, see [181]. However, eyes only capture visual stimuli during *fixations* that stand 200-400 ms. During this phase, the eyes stay stationary in the ROI. The fixation follows two oculomotor functions: rotation of the eyes as such the ROI falls on the two eyes' fovea, and then optimize the crystalline lens adjustment so that the retinal images become sharp [121]. Moreover, the image needs to be updated within 5 ms of fixation; otherwise, the observer may detect the low-resolution image due to foveated degradation [134, 38].

3.4. Eye tracking

Eye-tracking is a technique that detects user's eyes and calculates where or what they are looking at. The point where the user is looking is referred to as the *gaze point*. Modern eye trackers mainly rely on an infrared light source and video cameras to track black pupil circles and the white corneal glint, which is a projection of infrared rays from the outer surface of the cornea. During eye movement, the pupil follows the gaze direction, while the corneal reflection remains unaffected. The camera-based eye tracking systems can be categorized as near-eye vs. remote, on-axis vs. off-axis, model vs. regression-based, single vs. multi-camera input (see [110]). Duchowski et al. [215] classify gaze tracking into active, passive, single, and multi-modal. Besides, the *accuracy* of eye trackers is defined as the average distance between the real-stimuli position and the measured gaze position [16, 142].

3.5. Latency requirement

The higher precision and lower latency are of utmost importance for an optimized foveated rendering. Higher latency increases discomfort (i.e., simulation sickness, fatigue), perceptional degradation visibility, and artifacts [242]. The motion-to-photon (MTP) delay, a.k.a., end-to-end latency, consists of tracking latency and frame latency; defined as the time between capturing an eye/gaze movement and the frame reflection associated with the display change. The frame reflection is the duration between the GPU and the display, is generally half of the MTP delay. In modern graphics pipelines, 5 ms or less frame latency for stereo VR and 16-33 ms for gaming PCs are achievable [105].

Guenter et al. [80] suggest that VR has an optimal latency of 23 ms or less, but 40 ms or more is a delayed latency. Similarly, Albert et al. [8] recommend 20-40 ms as the most suitable value for latency for VR, while 50-70 ms is somehow tolerable, and 80-150 ms or more is unacceptable. On the contrary, Stengel et al. [203] report, 50-91 ms is the tolerable threshold. Li et al. [129] strongly suggest, for foveated rendering, the MTP delay should be less than 50 ms. Likewise, Stengel et al. [204] recommend the latency should never exceed 60 ms. Arabadzhiyska et al. [14] report that HVS sensitivity is fully restored within 40-60 ms after the saccade ends. Therefore, the frame should be updated within that time frame. In contrast, other authors [38, 105] report that the image should be updated within 5 ms after a saccade to avoid artifacts. Romer et al. [189] also suggest, for 360° video streaming, the latency should be approximately 20 ms. Similarly, Koskela et al. [118] report the latency for immersive applications should be less than 20 ms, which is further supported by the experiment [116] that use 14 ms latency under VR. However, Patney et al. [173] use 20-37 ms tracking latency in addition to the frame latency in their experiment. The figure 4 shows an overview of MTP delay which has been observed in multiple studies for VR applications.

Besides the *MTP delay*, pixel-row-update adds a considerable amount of latency to the desktop monitor. In the early 1990s, the MTP delay of 100-150 ms was more common for volumetric visualization [127]. However, the recent progress of processing power can remarkably lower that latency. Thunström's [218] study suggests that up to 42 ms latency is tolerable for 95% of the subjective studies with desktop monitors, whereas Loschky et al. [135] report 60 ms should be the standard. To sum up, for immersion, the best MTP delay should be < 5 ms, and on average ≤ 20 ms. Moreover, the MTP delay should never cross 50 ms regardless of display technologies. In addition, researchers have determined that the peripheral degradation at longer latencies (80-150 ms) must be reduced with respect to the amount considered acceptable at shorter latencies (50-70 ms), since the additional latency increases the likelihood of the viewer noticing visual artifacts in the peripheral area [135].

3.6. Visual acuity

Visual acuity or clarity of vision is described either as the Snellen value or Minimum Angle of Resolution (MAR). The normal visual acuity is defined as 20/20 Snellen value, equivalent to 1 arc minute in MAR in the

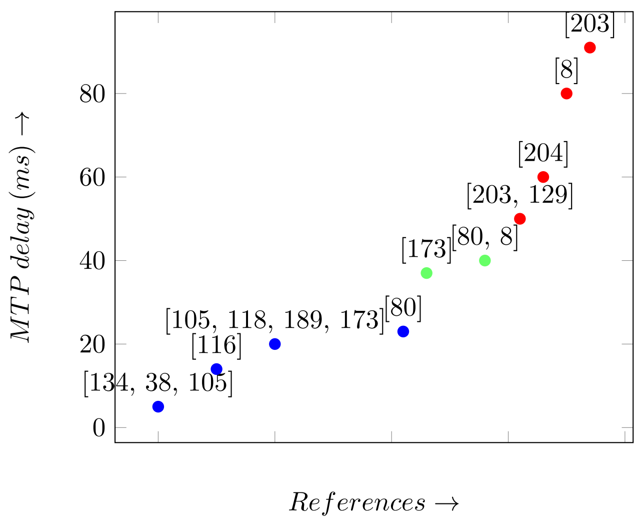

Fig. 4. Overview of Motion-to-photon (MTP) observed in different studies on VR applications. The studies are arranged from left to right from the smallest to the largest observed delay. In blue (5-20ms), we depict the best cases, in green the average case (around 40ms), and in red the worst cases (above 50ms). These thresholds are the average suggested thresholds indicated in the literature.

fovea [204, 206, 85]. Current foveated rendering research considers this normal visual acuity as a standard for display design. However, since, in realty, average viewers can barely achieve half of the maximum visual acuity, the most readable visual contents are designed for visual acuity of 20/40. Spjut et al. [102] suggest that 20/40 visual acuity should also be the standard for foveated rendered displays. However, Behnam et al. [22] report that commercial VR displays at the time of their survey (2017) hardly provide 20/90 visual acuity.

Studies tried to establish a relation (see Equation 1) of visual acuity fall-off from the visual axis [80]:

$$\omega_{cpd} = \omega_0 + m \cdot e \tag{1}$$

Here, ω_0, e, and m denote the smallest resolvable angle in cycle per degree (cpd), eccentricity in degrees, and slope respectively. The MAR model has been shown to fit low-level vision task findings as well as anatomical characteristics of the retina. Inverting the visual acuity results in the MAR as a linear model [245]. The minimum discernible MAR increases linearly with eccentricity $20° - 30°$ [203, 80, 243, 238]. However, according to few other studies, e.g., [206, 36, 67] visual acuity is subject to *hyperbolic fall-off*.

3.7. Contrast sensitivity function (CSF)

Unlike visual acuity, contrast sensitivity (CS) characterizes different aspects of visual function. Clinical trials often do not include CS in addition to visual acuity tests. Contrast is a difference in luminance, typically the difference in reflected light levels between adjacent points. CS function (CSF), expressed in cpd units, refers to the number of samples that can be discerned at a particular distance from the foveation point. It is defined as the reciprocal of the minimum contrast threshold (CT) to perceive a sinusoid of spatial frequency *f*, at different eccentricities *e* [72, 198, 260] (see Equation 2):

$$CS(f, e) = \frac{1}{CT(f, e)} \tag{2}$$

Humans can perceive with a resolution of 60-65 cpd in the fovea [200, 169, 7], gratings as fine as 1 arc-minute per pixel [216, 8] or equivalent of 120 pixels per degree (ppd) [98]. Interestingly, Cuervo et al. [50] report that humans with corrected vision have better than normal vision and the visual acuity ranges between 0.3-1 arc-minute (≈60-200 ppd). However, clinically 30 cpd has been considered as standard [110, 247].

Researchers have different views about the visual sensitivity fall-off with eccentricity. Weier et al. [244] suggest acuity reduces by 75% at an eccentricity 6°, whereas few studies recommend, after 20°, the sensitivity is reduced ten times [201, 143]. Similarly, Watson et al. [231] suggest that by 20° eccentricity, the human visual system can no longer resolve gratings narrower than 7.5 arc-minute per pixel. According to Akşit et al. [7], after 35°, the angular resolution drops to about 2.5 cpd, although Reddy et al. [184] recommend, the minimum visual acuity humans can perceive in the periphery is 8 cpd.

3.8. Adaptation effects

It is often reported that the HVS is sensitive to contrast in luminance ranging from $10^{-6} cd/m^2$ (objects viewed under illumination from the stars) to $108 cd/m^2$ (objects viewed on a bright sunny day) [186]. However, the instantaneous dynamic range is much lower, as it is limited to 4 orders of magnitude, with lower luminance perceived as noise, and higher luminance as over saturated uniform areas [146]. This is because humans extend their dynamic range by adapting to changes in the ambient luminance by moving as detailed vision windows along the luminance axis. Interestingly, adaptation is performed according to the luminance perceived in an area covering about one degree around the gaze direction, which is, however, frequently changing, also because of saccades [86]. Since the process of luminance adaptation is slower than gaze direction changes, as noted by Mantiuk et al. [146], in most situations the HVS is permanently in a maladaptation state.

4. Overview and classification

As discussed in detail in Sec. 3, the fovea centralis captures finer details than those captured in the periphery. By exploiting this, foveated rendering techniques achieve optimization by nonuniformly distributing the rendering effort, in particular by lowering the rendering fidelity in noncentral areas.

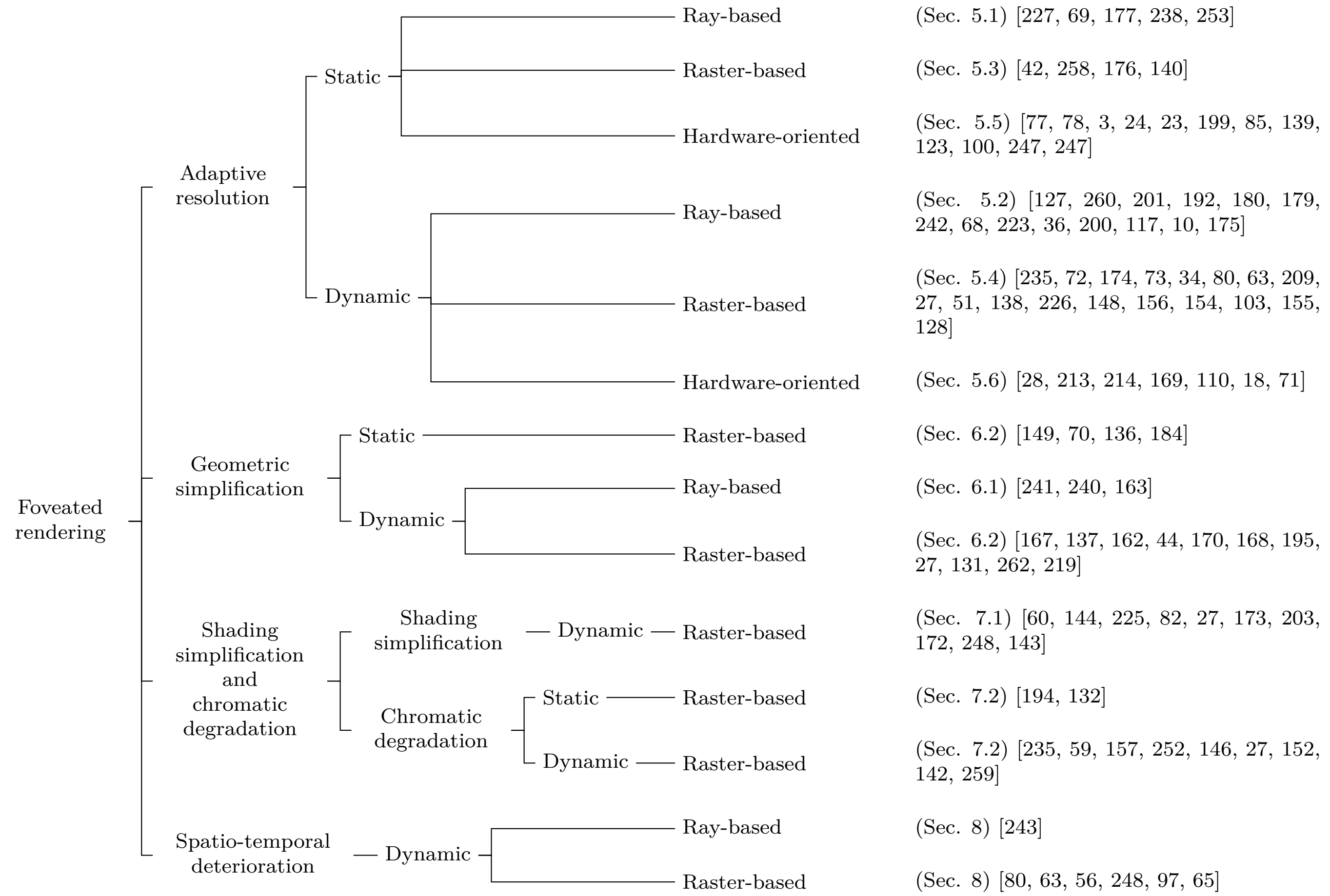

Fig. 5. The overall landscape of foveated rendering techniques (Sec. 5–8). The table focuses on methods, while applications are discussed distinctly in Sec. 9

. Each cited reference is assigned to the main class of technique. We further differentiate on whether it was originally applied for a static or dynamic gaze tracking and implemented for a ray tracing or ray casting pipeline.

Researchers have classified the foveated techniques in different categories, e.g., *experimental cognitive, algorithmic*, and *hardware approach* [90]. Regarding peripheral degeneration, Watson et al. [233, 235] recommend *geometric model, lighting-shading, texture*, and *window different resolution.* Accordingly, Swafford et al. [210] suggest four possible quality degradations in periphery: *resolution, screen-space ambient occlusion, tessellation*, and *ray-casting steps.* Similarly, Arabadzhiyska et al. [14] propose *spatial resolution, level of detail*, and *color* can be reduced in the periphery. Wang et al. [229] report that *geometry simplification, filter*, and *multi-resolution* can be applied to the periphery; however, this study is limited to video compression.

Our classification depicted in Figure 5 strives to seek commonality among rendering approaches. The main differentiation is among the types of degradation that are performed (Sec. 4.1). For each of these main classes, we further differentiate on whether the technique was originally proposed for a situation in which the gaze was assumed static or dynamic (Sec. 4.2). Finally, we also differentiate on whether the technique was originally implemented for a ray-based or a raster-based pipeline (Sec. 4.3).

In the following, we first provide general information on this classification. In the following sections, we will build on our classification to provide an in-depth analysis of the various methods that have been proposed in the literature.

4.1. Main classes of peripheral degradation

From a rendering method point of view, the fundamental differentiation is the type of adaptation that is performed. On this basis, we classify foveated rendering into four groups, depending on the type of peripheral degradation that is performed:

- *adaptive resolution* techniques work mainly in image space to reduce image density in the periphery (Sec. 5); these techniques include general-purpose approaches, as well as techniques tightly bound to specific display designs (*called Hardware-oriented* in this survey);
- *geometric simplification* techniques work instead, in model space, by adapting the complexity of rendered 3D models contributing to different areas of the display (Sec. 6);

- *shading* simplification and chromatic degradation techniques reduce, by contrast, the work per pixel, simplifying the quality of illumination simulation or chromatic fidelity (Sec. 7);
- *spatio-temporal deterioration*, finally, improves performance by adapting the refresh rate of pixels across the image, eventually reusing information from previous frames for less important areas of the display (Sec. 8).

4.2. Static versus dynamic gaze point

Independently from the type of peripheral degradation employed, foveated rendering assumes that there is knowledge of the gaze point, which determines how the effort has to be distributed across the image. While the specific type of solution used for obtaining this knowledge is not of primary importance for the rendering methods, some differentiation may exist among techniques that assume a static gaze point (e.g., at the center of the display), or techniques where the gaze point may dynamically vary across frames (e.g., on the basis of eye-tracking or other side information). For this reason, we distinguish between methods using a static gaze point and methods using a dynamic one. While we classify the presented techniques based on the setting in which they were originally introduced, some of the static ones may adapt to dynamic settings, and vice-versa, with few adaptations. We will point out these situations during the discussion. Nonetheless, presenting this classification also provides a view of the landscape of foveated rendering that shows the relative importance, and the historical evolution of static and dynamic setups.

Static foveated rendering schemes attempt to perform perceptual optimization without any additional tracking device. However, without gaze tracker, the typical assumption is that the user is looking at the center, and that degradation might be applied at the periphery of the image [183]. However, this is not a full-fledged technique, as human vision simultaneously involves *saccades* and *fixations* of the scene. Nonetheless, static foveated rendering techniques are still in widespread use, since they can be applied in a wide range of situations. Commercial near-eye displays, e.g., Oculus Rift and StarVR, have adopted this idea, sampling different regions with variable rates. It should be noted that this variable-rate sampling is also important to optimize rendering performance for these displays, as it leads to throwaway peripheral pixels hardly visible due to pincushion distortion.

On the other hand, more and more foveated rendering schemes take into account a dynamic variation of gaze. While methods relying on dynamic gaze variation can be used without trackers, e.g., by assuming that the viewer is following a particularly salient object, the large majority of these schemes are developed in conjunction with some tracking technology. Matthews et al. [150] differentiate eye tracking from gaze tracking by stating that eye tracking only measures eye movement, while gaze tracking tracks the observer's head position to determine the actual gaze point in the virtual world. In our work, we are not making fine differentiations, as we are interested in how the gaze position is exploited by optimized rendering algorithms. For this reason, we cover a wide range of trackers, e.g., *position tracker, optical tracker, face tracker* under the *gaze tracker umbrella term.*

Depending on the purpose, the latest tracking hardware has either high accuracy and lower update frequency, or vice versa. Studies suggest that for optimizing foveated rendering, high frequency is more significant than high accuracy [16, 80]. Although few studies suggest that head tracking is adequate for noncritical purposes, as the human eye focuses closely on the head orientation ($\pm 15°$ radius [20]), Lawrence et al. [64] report that, in VR applications, inaccuracies and latencies may lead to motion sickness and nausea. For this reason, applications like *immersion, cloud-based gaming* explicitly require accurate and low latency tracking; hence, for foveated rendering an eye tracker appears as the best option for such cases. However, due to the viewing distance and relative motion between the viewer and the display, eye-tracking is inconvenient for large high-resolution display walls [26]. As an alternative, for such setups, *position* and *optical-tracker* give an approximate gaze position considering the observer's FOV with higher latency.

4.3. Ray-based versus raster-based techniques

Finally, the implementation of the degradation techniques may also vary depending on the rendering pipeline employed. In particular, while a large variety of combinations exist, ray-based techniques make it simpler to perform per-pixel adaptations, raster-based techniques typically favor model-space solutions. We, therefore, differentiate between ray-based and raster-based methods. Ray-based and raster-based pipelines are directly available on modern programmable graphics hardware. Since foveated rendering requires real-time graphics, several pipeline-specific approaches have been implemented.

Typically, in foveated rasterization, the gaze point may be used to select geometric levels of detail for the displayed models, as well as an input for a fragment shader. The shader code will run a simplified fragment if it detects that the user is not looking at the current target pixel. These approaches make foveation easy to integrate with rasterization pipelines. Complications, however, arise from the implementation of realistic shadows, reflection, refraction, caustic effects, and global lighting, which often require the tuning of shadow mapping, reflection mapping, and other rendering techniques to cope with variable-resolution rendering [68].

On the other hand, the ray-based approaches are better applicable to photo-realistic graphics rendering since the path of the rays is computed pixel by pixel [243]. Optimization for foveated rendering is most often achieved by reducing the number of rays in non-foveal areas. Ray-based techniques have shown the ability to easily simulate complex illumination patterns, but, in real-time settings, such techniques require important resources, especially for dynamic objects, due to the need of recomputing spatial indexing to achieve logarithmic complexity [160]. Foveation has shown to be an effective optimization technique due to the massive potential reduction in the number of rays. For foveated path tracing, Koskela et al. [118] provided a theoretical estimation of performance gains available and calculated that 94% of the path rays can be omitted. For this reason, they identified foveated rendering as an essential technique to use path tracing within VR applications.

With the evolution of graphics pipelines, however, the boundary between rasterization and ray tracing is becoming more and more blurred. Ray-casting or even ray-tracing may be performed in fragment shaders, while rasterization is often used for the view rays in a ray tracing or path tracing solution. In our classification, we will, nonetheless, conserve this distinction by presenting the various techniques in the setting in which they were originally introduced, eventually cross-linking similar ray tracing and rasterization techniques. By doing so, we aim to provide a view of the evolving landscape of foveated rendering implementation frameworks.

5. Adaptive resolution

The *first group* of methods in our classification (Figure 5) strives to reduce the peripheral resolution to accelerate the rendering process. This is the most common approach in foveated rendering. Over time a wide range of techniques has been developed, such as *adaptive sampling mask, multi-resolution pyramid, discrete cosine transform (DCT), wavelet transform, log-polar transform, log-rectilinear transform.* The adaptive resolution is applicable on CPU, GPU, and even on a hybrid architecture. Besides, both ray-based and rasterize graphics pipelines have been used to reduce resolution, few techniques even combine both pipelines. Unconventional approaches include mostly dual display setup, e.g., *inset-based projection, overlapped region,* and *focus plus context.* Recent progress of AR displays, especially holographic, varifocal, and light field displays rely heavily on the adaptive resolution to reduce rendering load. However, *flickering, pop-up* and other visual artifacts are often visible that require additional postprocessing.

In this section, we will survey the adaptive resolution techniques according to our classification. A general overview of the surveyed methods is presented in Table 1 for the general-purpose techniques, and in Table 3 for the methods tightly bound to a specific hardware setup. In the following, we will first discuss each of the subclasses (see Sec. 5.1–5.6), before summarizing our findings (Sec. 5.7).

5.1. Static ray-based techniques

Ray-based rendering techniques (see Table 1), such as ray tracing, path tracing, and ray casting, are well adapted to foveated rendering because of the adaptive sampling control over the frame, high-quality shadows, reflections, refraction, translucency, caustic effects, and other visual qualities.

Static real-time *foveated ray tracing* systems reduce spatial sampling by imitating the human non-uniform and sparse vision characteristics, typically assuming that the viewer is looking at the center of the display. This approach is often used for near-eye displays. Fujita and Harada [69], for instance, developed a foveated ray tracer for a headset, in which the sampling pattern is distributed with $\theta^{-2/3}$, where θ is the angular distance from the display center. To avoid artifacts due to sparse sampling, pixel colors are computed by averaging a set of neighboring samples in the image plane.

Pohl et al. [177], in a head-mounted display (HMD), combined density reduction due to foveation with the fact that lenses in modern consumer HMDs introduce distortions like astigmatism, in which only the center area of the displayed content can be perceived sharp while, with increasing distance from the center, the image gets increasingly blurred. This reduction is encoded in display-specific precomputed static sampling maps, which are images that encode the number of sampling rays per pixel (255 being the maximum of allowed supersampling). Moreover, they achieve considerable speed-up by combining density control with image quality control. In particular, in addition to lowering density inside areas, they employ high-fidelity CPU ray tracing in the display center, and faster GPU-accelerated rasterization in the periphery. Moreover, pixels that are very far from the center are not rendered upon head motion, reusing pixels from previous frames to avoid illumination changes [178]. This hybrid technique significantly improves the graphics quality at higher frame rates: with user-specific calibration, the demonstrated rendering speedup reached up to 77% on several benchmark scenes. This method was later extended to dynamic gaze tracking using an eye tracker [179] (Sec. 5.4). Recently, Yang et al. [253] varied the ray tracing rate based on scene specific information to reduce the number of shading samples. The particular use case is the usage of path tracing for computing illumination in a deferred shading pipeline. Sample rate is reduced by combining a foveation terms with terms depending on BRDF complexity and distance to viewer. Results demonstrate speed-ups of up to 30%.

Static foveation has also been used for other types of displays. For instance, Wei and Sakamoto [238] use density reduction to optimize rendering speed for an experimental holographic display. Such a display technology simulates the recording part of traditional optical holography by using a computer, saving light information as electronic data called an interference pattern. This approach, however, requires a large amount of calculation. Therefore, for foveated rendering, instead of adapting pixel density, they reduce the angular resolution of these calculations depending on the distance from the look-at point, assumed at the center of their display. Only the area within 5° to the center is rendered at full resolution, while the rest (up to 8° on their experimental display) uses a lower angular sam-

Table 1. Summary of different techniques developed to achieve adaptive resolution, similar approaches are grouped together. Methods tightly bound to a specific hardware setup are presented separately in Table 3

Algorithm used	References	Static	Dynamic		Pipeline	
			Eye-tracker	Gaze-tracker	Ray-based	Raster-based
Adaptive ray tracing	Fujita and Harada [69], Wei and Sakamoto [238], Yang et al. [253]	●	○	○	●	○
Adaptive ray tracing	Levoy and Whitaker [127], Siekawa et al. [201], Peuhkurinen and Mikkonen [175]	○	●	○	●	○
GPU-accelerated ray tracing	Weier et al. [242], Siekawa et al. [200]	○	●	○	●	○
Luminance aware rendering	Tursun et al. [223]	○	●	○	●	●
Dynamic sampling map	Pohl et al. [179]	○	●	○	●	○
Hybrid approach	Pohl et al. [177]	●	○	○	●	●
Hybrid approach	Pohl et al. [180], Friston et al. [68]	○	●	○	●	●
Adaptive path tracing	Roth et al. [192]	○	○	*Head*	●	○
Path tracing in log-polar space	Koskela et al. [117]	○	●	○	●	○
Adaptive ray casting	Viola et al. [227]	●	○	○	●	○
Adaptive ray casting	Zhang et al. [260], Bruder et al. [36]	○	●	○	●	○
Adaptive ray casting	Ananpiriyakul et al. [10]	○	○	*Face*	●	○

Head = head-tracker *Face = face-tracker*

pling rate. The static setup makes it possible to exploit the precomputation of sampling patterns.

5.2. Dynamic ray-based techniques

Dynamic techniques receive new gaze information at each frame and must update the display with low latency.

The *first group* of techniques in this area is purely ray-based and achieves optimization by reducing the number of rays and reconstructing images from sparse samples. Levoy and Whitaker [127] developed the earlier *volumetric rendering* with adaptive ray tracing, getting the gaze points with an eye tracker. In the algorithm, depending on the distance to the gaze point, three regions of a scene are gradually sampled at $1, 1/2$, and $1/4$ of the native resolution and then blended for generating a continuous final rendered image. Similar approaches have been later used for *Whitted-style* ray tracing of simple scenes [201].

With the introduction of programmable graphics pipelines, several more elaborate approaches were introduced, with the goal of having a finer control of ray generation and reducing artifacts, especially at the periphery. Siekawa et al. [200] use GPU-accelerated ray tracing with four different sampling masks for a nonuniform distributed set of pixels to reduce the number of traced rays. To reduce flickering artifacts in the periphery, which is very coarsely sampled, strong temporal anti-aliasing (TAA) is applied. Peuhkurinen and Mikkonen [175], instead, distributed rays according to a log-polar transformation rather than discrete masks and demonstrated ray-tracing for simple sinces on mixed reality application. Likewise, Weier et al. [242] combine GPU-accelerated ray tracing with a depth of field filter (DOF). The ray-tracing step in the algorithm samples the image sparsely based on a visual acuity model, and then the temporal stability of peripheral image regions is enhanced using reprojection-based TAA. Finally, the complete image is computed from sparse samples using pull-push interpolation, and gaze-contingent DOF is computed as postprocessing. Although the model was originally developed for foveated artifact reduction, it also reduces shaded samples up to 70%.

Tursun et al. [223] noted that, while previous foveated solutions reduce resolution purely as a function of eccentricity, human visual sensitivity is also strongly influenced by the displayed content. They thus studied the resolution requirements at different eccentricities as a function of luminance patterns, deriving a low-cost parameterized model. The model is used in a multipass rendering technique, which predicts the parameters from a low-resolution version of the current frame. As a result, the model proved to be capable, on benchmark scenes, to use only 47% of the rays to render the foveated region, without visual artifacts like pop-up effects and tunnel vision. For further speed-up, variable-rate shading [205], which distributes shading samples over time, is also employed. The overall approach benefits from the flexibility of the CUDA block-wise architecture.

The *second category* of algorithms is *hybrid approach* in which both ray tracing and rasterization have been combined for faster computation. Pohl et al. [180] noted that when the user is not looking at the center of a head-mounted display, not all of the image is seen. User tests showed that, in their particular configuration, on average, 57% pixels were typically invisible in the entire frame. In a fully ray-traced pipeline, they skipped rays detected as invisible, while in a rasterization pipeline the invisible pixels are stenciled out, avoiding shading computation. The study was then extended by combining rasterization at the periphery with ray-tracing at the center [179], also including dynamic sampling maps and lens astigmatism [178]. For performance reasons, dynamic sampling maps are recomputed per frame depending on the current gaze at a low resolution and interpolated to get the required amount of rays per pixel. Taking into account the gaze point resulted in a speedup of 20% with respect to the static solution.

Since multipass approaches are prone to introduce latency, Friston et al. [68] introduced a single-pass rendering technique based on a single perceptual rasterization pass. Their approach combines two solutions. First of all, they implement rasterization into a frame buffer with a non-constant pixel density that peaks at the fovea. Each rasterized pixel computes illumination with ray tracing. Second, they update every column of pixels at different times. The latter feature can be used on HMDs with rolling displays, such as Oculus Rift DK2, that illuminate different spatial locations at different times. As a result, they achieve a performance similar to warping solutions, without the limitations with respect to disocclusions, object motion, and view-dependent shading, while reducing the aliasing artifacts of foveated techniques based on sparse ray sampling at every frame.

A number of approaches generalize the above concepts to *foveated path tracing*, the *third category* in our classification, in which performance gains are achieved by controlling the shading complexity through the reduction in a number of traced paths. As for the typical real-time path-tracing solution, the final image is generated by a denoising filter from the noisy result of path tracing. A notable approach has been proposed by Roth et al. [192], based on the NVIDIA OptiX framework. Their implementation targets high-resolution displays in which the user's FOV is precalculated, with more dense rays traced in the fovea region, and sparser rays traced in the periphery, where a Gaussian filter is also applied to blur the image to mask aliasing problems. This third category is currently less explored, mainly due to the difficulty of computing global illumination in a very time-constrained setting with strict latency bounds. A recent study by Koskela et al. [117] implemented real-time path tracing in log-polar space. In their benchmarks, both rendering and denoising achieved a 2.5× in a VR setup. However, jittering effects could be observed in both the fovea and periphery.

The *fourth set* of techniques is based on *foveated ray casting* commonly used to render massive 3D models or volumes. Ray casting is used here due to its flexibility and efficiency in visibility computation in combination with precomputed acceleration structures. The techniques used in this area do not significantly differ from the previously discussed solutions. Zhang et al. [260] present real-time foveated ray casting base on adaptive sampling mask and CSF with significant frame-rate improvement. Similarly, Bruder et al. [36] develop ray casting technique derived from Linde Buzo Gray sampling [54] and natural neighbor interpolation that leverages visual acuity fall-off to speed up volume rendering. Without any perceptible changes in visual quality, this technique achieved speed up to 3.2 fold on the presented benchmarks. Likewise, Ananpiriyakul [10] apply adaptive ray casting on vector and volume visualization in which the step size increases along with eccentricity, resulting in faster computation and interaction latency decline. Interestingly, the approach uses a face-tracker instead of conventional gaze-trackers.

Dynamic ray-based techniques for adaptive resolution are a well-researched and still very active area, where most of the literature in the 2014-2020 time frame were produced. This is because these techniques make it natural to finely and rapidly adapt sampling rates based on eccentricity and other measures. However, due to decreased ray density, artifacts like flickering are often visible in the periphery. Therefore, additional postprocessing, e.g., strong antialiasing [200, 242], and denoising [242] are essential.

5.3. Static raster-based techniques

Rasterization based techniques produce images by projecting the scenes on a regular grid. This regularity is exploited by several foveation methods to design specialized adaptive sampling and reconstruction techniques (see Table 2).

The *wavelet transformation* is, in particular, at the root of the major rasterization-specific approaches to foveation. In the wavelet domain, the images are decomposed into different components and frequencies [141] in which each level can represent the different scales of information. In the context of foveation, wavelet representations are often used to control the sampling rate both in image space, to control the number of samples, and in object space, to control the sampling. In particular, variable resolution for foveated volumetric representations can be achieved by controlling the number of wavelet coefficients. Chang et al. [42] employ the Gaussian smoothing function as an integral operator and analyze its kernel for achieving space-variant degradation. Piccand et al. [176] develop volume data visualization technique based on 3D Haar wavelet transformation. In this approach, the ROI is rendered at full resolution, while contextual areas at coarser resolution are rendered through wavelet splatting. One main drawback of this method is that the contextual region pixelates due to the combination of Haar wavelets with splatting. Yu et al. [258] render volume data using wavelet coefficients under selected tracked rays. This is a two-step process: rapid reconstruction of the super-voxels from wavelet coefficients, and then render the super-voxels by tracking rays with different thicknesses. To reduce staircase artifacts, a space-variant smoothing filter is applied.

Variable spatial resolution is also achieved by using standard rasterizers with different configurations in the various areas of the screen. A prominent example is the rendering framework proposed by Malkin et al. [140], that assembles the final image from square fragments rendered separately, each of which has been blurred according to the distance from its midpoint to the point of fixation. Such a decompo-

Table 2. Summary of different raster-based techniques developed to achieve adaptive resolution, similar approaches are grouped together.

Algorithm used	References	Static	Dynamic		Pipeline	
			Eye-tracker	Gaze-tracker	Ray-based	Raster-based
Wavelet transformation	Chang et al. [42], Yu et al. [258], Piccand et al. [176]	●	○	○	○	●
CUDA opt. architecture	Malkin et al. [140]	●	○	○	○	●
Adaptive sampling	Vieri et al [226]	○	●	○	○	●
Adaptive sampling (3 layer)	Guenter et al. [80], Finch et al. [63], Marianos [148]	○	●	○	○	●
Adaptive sampling	Cuervo and Chu [51]	○	○	*Head*	○	●
Adaptive sampling (2 layer)	Swafford et al. [209], Bektas et al. [27], Lungaro and Tollmar [138]	○	●	○	○	●
Adaptive sampling (2 layer)	Watson et al. [235]	○	○	*Mouse*	○	●
Multi-layer pyramid	Perry and Geisler [72, 174, 73]	○	●	*Mouse*	○	●
Spatiotemporal filtering	Bohme et al. [34]	○	●	○	○	●
Log-polar transform	Meng et al. [156, 154, 155]	○	●	○	○	●
Log-rectilinear transform	Li et al. [128]	○	●	○	○	●

Head = head-tracker *holo = holographic display*

sition into tiles allows for an efficient parallel CUDA-based implementation.

Rasterization-based techniques are also often used in conjunction with nonconventional display setups, such as near-eye displays or light-field display. Since most of these methods have been specifically designed to take into account display-specific features, they are described in a separate section on Hardware-oriented techniques (Sec. 5.5).

5.4. Dynamic raster-based techniques

The most explored foveated rendering research area comprises dynamic raster-based techniques that vary local image resolution in response to gaze changes (see Table 2). Due to the need for low-latency and high frequency display, these techniques must employ several optimization schemes that permit fast adaptivity in conjunction with moving ROIs. In this section, we discuss sub-sampling [80, 63, 148, 226, 51, 209], multi-layer pyramid [72, 174, 73, 34], and log-polar transformation [156, 154, 250, 128] which are used to achieve adaptive resolution.

The *first set* of techniques is based on *compositing different resolution images* to quickly produce a foveated display. The most classic technique is to use a multi-pass approach, in which several image layers around the tracked gaze point are rendered at progressively higher angular size but lower sampling rate, and then rescaled and composited to produce the final multi-resolution image. For instance, Guenter et al. [80] introduced a multipass rasterization pipeline for 3D graphics based on the acuity fall-off model proposed by Levoy et al. [127], in which the scene is rendered on three nested and overlapping render targets centered around the current gaze point. The inner layer is smallest in angular diameter and rendered at the native display resolution, while the two peripheral layers cover a progressively larger angular diameter but are rendered at a progressively lower resolution and bilinearly upsampled before merging them with the others. Note that this system also used coarser scene LODs for peripheral layers (see Sec. 6) and updated them at half the temporal rate (see Sec. 8). Through this approach, half of the shading cost was saved with a 5-6 times overall graphics performance improvement demonstrated on a desktop HD display. The system was later extended for a 3×3 tiled LCDs, demonstrating up to 10-15 times less rendering cost with 6-8 times average speedup [63]. The reduction in the density of peripheral layers leads to distracting strobing and crawling artifacts and makes anti-aliasing based on super-sampling harder. For this reason, the cost of anti-aliasing is also amortized over multiple frames, using a combination of multisample antialiasing (MSAA), temporal reverse reprojection [165], and temporal jitter of the spatial sampling grid [49].

Many follow-ups used the same architecture. For instance, Marinos [148] use three layers: 100%, 60%, and 40% resolution which depends on the Euclidean distance from ROI. Likewise, Cuervo and Chu [51] investigate the panoramic stereo video and likelihood-based features in which the video is subdivided into three regions: high, medium, and low resolution. An integrated convex-like optimizer adapts to real-time head movement and reallocates pixels according to the motion. In contrast, instead of three layers, Swafford et al. [209] use two sample layers, full resolution in the fovea and 25% resolution in the periphery. Lungaro and Tollmar [138] also employ dual resolution on the video delivery framework by applying an optimized foveal mask to each frame. Such a 2-layer architecture was also used in early user studies [235] that demonstrated that lowering resolution in the periphery of HMDs did not affect user performance on complex visual search tasks. This multiple-image rendering architecture is also used to drive recent VR displays, e.g., the very high-resolution display by Vieri et al. [226], a 4.3" OLED display with 18 *megapixels/eye*, and 120 Hz refresh rate.

Since reducing resolution is prone to introduce visible artifacts, other authors have presented architectures that improve image quality by supporting compositing and filtering of multiple images. The *second group* of methods is used to create a space-variant resolution to the periphery, is known as *Multi-Layer Pyramid (MLP).* Geisler et al. [72] combine CSF with MLP for faster *video communication* over low bandwidth networks. In this procedure, the entire scene is divided into six levels, and each level is then motion-compensated, multi-resolution coded, and quantized based on HVS. Finally, lossless encoding and foveated video quality assessment metrics have been integrated into foveated compression algorithm. Similarly, Perry and Geisler [174, 73] use MLP with filtering at each pixel location, achieved by interpolation between levels of the pyramid using the resolution map. Derived from the gaze-directed spatial resolution function developed by Perry and Geisler [174, 73], Böhme et al. [34] employ a gaze-contingent spatiotemporal filtering technique that uses a *resolution map* to specify the optimal temporal resolution at the ROI. As a result, the authors claim smooth, and artifact-free real-world video output.

While the above techniques partition the image into a small set of discrete areas that are then composited, an alternative approach is to directly produce a seamless variable-rate image by warping the angular distribution. The *third category* of algorithms is based on *logarithmic transformation.* Meng et al. [156] develop the kernel foveated rendering (KFR) technique in *log-polar coordinate.* In the method, first, a log-polar transformation has applied in the buffer memory, and then inverse log-polar transformation with anti-aliasing has applied to reduce the resolution. However, in the presented benchmarks, the technique achieves 2.0-2.8 times speedup for 3D texture meshes and 2.9-3.2 fold better performance than *ray casting* rendering on a 4K-UHD. In an extension, Meng et al. [155] use *eye dominant* feature that implements a lower foveation rate for the dominant eye than the non-dominant. In comparison with KFR [156], an additional 1.06-1.47 times speedup was achieved. In another study, Meng et al. [154] extend the KFR to 3D light field display. The 3D-KFR is parameter-dependent, embedding polynomial kernel functions in the classic log-polar mapping. Nonetheless, there are two key research challenges in KFR methods, *first,* the user-dependent optimized parameters that make it difficult for practical implementation, and *second,* artifacts such as flickering are frequently visible. To reduce artifacts, Li et al. [128] use log-rectilinear foveated rendering. Results from this research prove that log-rectilinear transformation with *summed-area table* sampling against log-polar transformation effectively reduces flickering artifacts and saves bandwidth.

Other dynamic rasterization-based techniques have been also developed to take into account the special characteristics of nonconventional displays. Those methods are described in a separate section on Hardware-oriented techniques (Sec. 5.6.

5.5. Static hardware-oriented techniques

While the approaches discussed so far are general-purpose techniques for achieving variable resolution across images, several methods have been designed for particular displays with unconventional characteristics. These include, e.g., dual displays [77, 78, 3], varifocal displays [247, 256], and holographic displays [85, 139, 123, 100]. Here and in the following section, we cover such hardware-oriented approaches to achieve adaptive resolution, focusing in particular on how raster-based and ray-based techniqeus have been adapted to those configuration (see Table 3). In this section, we will first focus on static configurations with a fixed gaze point, while in the next we will cover the dynamic case.

The *first set* of techniques uses a *physical dual-display* setup to achieve variable resolution. A typical design is the earlier foveated dual display approaches, which were mainly inset-based, with higher resolution at the center and coarser resolution elsewhere. On these displays, rendering techniques typically need to perform two renderings and take into account continuity between the presented images. Godin et al. [77, 78] designed a dual-resolution foveated stereoscopic projection setup that superimposed images with opposing polarization that is suitable for exploring large models and environments consists of high geometric and texture complexity (the display setup targeted over 10 megapixels). However, there are few downsides, e.g., color, resolution, brightness variation, and the line between different projectors. Therefore, image warping is applied as a part of the rendering pipeline to overcome these challenges. Ahlborn et al. [3] introduce a multi-projector wall where the coarser-resolution is projected from a rear projector. To modify the OpenGL pipeline without modifying application code, they implemented the inset controller as a Chromium SPU. Another front projector with a mechanical *pan-tilt mirror* projects small though high-resolution images overlapped. Baudisch et al. [24, 23] develop a focus plus context (FPC) display in which foveation is possible during image acquisition. Besides, Shimizu [199] develops an advanced *wide-angle foveated* (AdWAF) model that uses an especial lens to distort the acquired image geometrically into four regions by combining both Cartesian and logarithmic coordinates. As compared to a log-polar model, the AdWAF model minimizes image data by more than 13%.

The *second set* of techniques is explicitly developed for near-eye image presentation. Sometimes, displays in this category are explicitly designed taking into account foveation in their design, but no particular rendering technique is required, besides taking into account the fixed variable angular resolution of the display. One example is the *varifocal AR display* of Wu and Kim [247], which

Table 3. Summary of different hardware-oriented techniques developed to achieve adaptive resolution, similar approaches are grouped together.

Algorithm used	References	Static	Dynamic		Pipeline	
			Eye-tracker	Gaze-tracker	Ray-based	Raster-based
Multi-layer point cloud (holo)	Hong et al. [85], Hong et al. [100]	●	○	○	○	●
Phase-only (holo)	Maimone et al. [139]	●	○	○	○	●
Multi-layer with PSF (holo)	Lee et al. [123]	●	○	○	○	●
Multi-layer (var)	Wu and Kim [247]	●	○	○	○	●
Geometric phase lens	Yoo et al. [256]	●	○	○	○	●
Dual projector	Godin et al. [77, 78], Staadt et al. [3]	●	○	○	○	●
Focus plus context	Baudisch et al. [24, 23]	●	○	○	○	●
Wide angle lens	Shimizu [199]	●	○	○	○	●
Electronic circuit board	Park et al. [169], Bae et al. [18]	○	●	○	○	●
Adaptive resolution (var)	Kim et al. [110]	○	●	○	○	●
Dual display	Benko et al. [28]	○	●	○	○	●
Dual display	Tan et al. [213, 214]	○	●	*PBPD*	○	●
Dual layer LCDs	Gao et al. [71]	○	●	○	○	●

PBPD = Pancharatnam-Berry Phase Deflector *var= varifocal display* *holo = holographic display*

allows retrofitting a medically prescribed lens with a varifocal lens for vision correction. Remarkably the prototype can achieve angular resolution up to 22 cpd for the virtual image at the center (6°) where the rest see-through display has a uniform 32 cpd resolution. Another typical example in this category is the near-eye display of Yoo et al. [256], which uses a fixed high resolution at the fovea and a lower resolution in the periphery, exploiting polarization-dependent doublet geometric phase lens and temporal polarization multiplexing methods to produce the images.

Holographic displays, with respect to standard binoculars, use wavefront modulation to offer full depth cues. These displays require large amounts of computation to compute the diffraction patterns, and using adaptive resolution is essential. The first set of solutions perform hologram synthesis in real-time from 3D point clouds using the *Rayleigh-Sommerfeld* diffraction formula. To achieve foveation, the data is represented as a multilayered point cloud, in which each layer has a different density according to MAR[85]. This model was then adapted to combine the holographic and two-dimensional displays to provide 3D images near the fovea and 2D images at the periphery [100]. Moreover, the point cloud is upsampled in the periphery to avoid holes. Maimone et al. [139] concentrated, instead of the design of a *phase-only* holographic projection with a spatial light modulator, showing how true 3D holograms can be generated directly from the output of the standard graphics pipeline through a post-processing step. In particular, they introduce a real-time computation method based on linearly separable convolution to achieve spatially variant focus and aberration correction for eye-tracked displays. The prerequisite for high-speed computation is a spatially invariant lens phase function, which implies that the focus and aberration correction is constant over the image. Foveation is exploited by providing the correct lens function where the user is looking rather than computing or approximating the full spatially variant solution.

Multi-layered displays, by contrast, can provide continuous focus cues within a working range by decomposing 3D scenes into 2D layer images, that can be presented through a variety of optical designs. Lee et al. [123] use for that purpose a light guide and a holographic lens. The major problem for such displays is to compute the layer images and computationally optimizing them to provide appropriate focus cues. Instead of using simple depth-weighted blended, per-image weights are optimized by comparing perceived retinal images with target retinal images according to the focal depth of the eyes. Foveation and eye movement are taken into account by minimizing the degradation of contrast within the fovea while considering a large eye box enlarging the eye box that takes into account possible eye movements. Contrast ratio curves and *visual differences (HDR-VDP2)* [145] are used for that purpose. The method has the drawback of being very sensitive to calibration and requires important computation resources, with the prototype achieving 10Hz for a 700×350 retinal image on an NVIDIA board.

5.6. Dynamic hardware-oriented techniques

A number of specialized hardware solutions to create displays that adapt resolution based on user's gaze. The *first set* of techniques is based on *physical dual displays*, complementing the dual display solutions presented in section 5.3 with components dedicated to dynamic gaze tracking. As for the static case, the only notable variations in terms of rendering algorithms are related to aspects needed to cope with particular display features. A typical example is given by Benko et al. [28], who couple a tracked optical see-through display with a projector-based spatial AR display. Their multipass approach renders the scene five times: twice for the glasses (once for each eye), once for the projected periphery, once for the projected inset, and once for the projection mapping and compositing process for the projector view. The projected inset renders occlusion shadows for the glasses content or only shows the surface shaded content that is not view-dependent. Visual discontinuities are reduced by applying a smooth transition between the periphery and the inset. Similar multi-pass rendering techniques can be applied to the display design of Tan et al. [213, 214], who achieve the realization multi-resolution foveated display panel with a combination of two separate OLED panels and a beam splitter which is used as an optical combiner. The first monitor has a wide FOV but low resolution, while the second display has super high resolution in the central region (25°). For dynamic foveation, a switchable liquid crystal-based *Pancharatnam-Berry Phase Deflector* is applied that shifts the high-resolution regions with contents. However, the *Pancharatnam-Berry Phase Deflector* can be replaced with an eye tracker. As for Benko et al. [28], each physical display is handled by a different rendering pass.

The *second set* of techniques is developed to achieve foveated resolution through *electronics circuit*. Park at al. [169] assumes that the renderer performs a vertical resolution reduction depending on the Euclidean distance from the gaze point, while keeping the horizontal resolution fixed. A specialized circuit using multiple line driving gate drivers then decompresses the image for display. Subjective assessment, PSNR, and SSIM indexes proved that the foveation-based driving scheme can be used without causing any noticeable deterioration. Since the display Rendering techniques must be aware of display resolution to suitably distribute pixel samples. Bae et al. [18] perform instead an adaptation in both horizontal and vertical direction by proposing a *variable clock generation circuit* to manipulate output waveforms of shift registers for OLED display. The electromagnetic circuit, which is made up of four thin-film transistors and one capacitor, generates pulses with variable widths that correspond to twelve resolutions in the display region. The above-mentioned rendering method can be directly employed to speed up the rendering for these variable resolution displays.

The *third group* of foveated techniques is designed for unconventional displays, such as *light field,* and *varifocal displays.* In contrast to conventional near-eye displays, these displays can create better visual cues and an immersive experience. Gao et al. [71] combine dual-layer LCDs and magnifying lenses to develop a *light field display.* In the system, a *Hadamard product* [112] of two-layer patterns is used to restore the light field scene. Besides, the LCDs need to be flipped vertically, and the optical distortions are calculated in post-processing. Kim et al. [110] design a state-of-the-art foveated *varifocal AR display* in which the resolution and focal depth cues are driven by eye-tracking. Besides, the display combines a traveling microdisplay, a concave half-mirror magnifier, and a laser projector-based Maxwellian-view display. Since the overlap between the fovea and periphery is visible, a *stencil mask* to the outer paths of the foveal image is used.

5.7. Discussion

Achieving adaptive resolution through foveated rendering is a wide research domain. One common use of foveated rendering is to subsample various regions of a scene to different resolutions and blend them. The number of layers used in various studies varies, for example, two [209, 51], three [80, 63, 148], and even six layers [72] have been used. Further, a distinct subsampling ratio also has been applied. However, the $1, 1/2, 1/4$ sampling rate for three layers by [127] have been widely adopted in [80, 238]. Nonetheless, these techniques are not free from artifacts like flickering and require strong *TAA* in post-processing. Among other algorithms, the wavelet transformation [42, 258, 176] suffers from sudden pixelation, and consequently smoothing filtering like Gaussian is required. Along with other aspects, the *log-polar transformation* [156, 155, 250] calculation is parameter-dependent, and a time-consuming user study is prerequisite for optimization.

Since ray-based methods allow arbitrary sampling patterns in screen space, foveated techniques can apply more easily than rasterization. Due to the GPU robustness and affordable price, the foveated ray tracing has gained much interest in recent years [69, 238, 201]. Moreover, the CUDA architecture that supports the implementation of both ray tracing [223], and rasterization [140] through general-purpose parallel programming techniques offers large flexibility. In recent years, the boundary between rasterization and ray tracing is becoming more and more blurred, and hybrid approaches are emerging [68, 177].

While foveation can be applied to standard displays, it is increasingly employed in conjunction with new technologies such as *varifocal, light field,* and *holographic displays.* There are several advantages of these displays; e.g. achieving continuous visual cues, and solutions for *vergence and accommodation conflict* that lead to fatigue for near-eye 2D displays with OLED/LCD. Among other advantages, the varifocal AR display can reach large FOVs (e.g., $85^\circ \times 78^\circ$) coupled with high angular resolution (e.g., 60 cpd angular resolution in the fovea [110, 247], while more traditional displays are typically much more limited (e.g., achieving a maximum 40° FOV and 10-15 cpd of angular resolution). However, several key research challenges exist in unconventional displays. In particular, most of the *holographic, varifocal, light field display* research is limited to static foveation, however, and dynamic foveation solutions have started to appear only recently [110, 103]. The rendering complexity for these displays (especially for holographic ones) is also very high, and most presented solutions are limited typically to simple scenes using simple shading models, most of the time demonstrated in standard rasterization pipelines (see table 1). Extending these displays to the photorealistic rendering of complex scenes is an open research challenge.

Dual-display setups are a very common solution found in foveated rendering. Projection-based dual display setups emerged as a viable solution to achieving higher resolution through projection on large screens. However, at present, this solution is being employed more and more frequently for near-eye displays, which use the technique to combine a large resolution at the fovea with a wide FOV (e.g., [28, 213, 214].

6. Geometric simplification

The *second group* of methods in our classification (Figure 5), instead of, or in addition to, reducing image resolutions, strives to improve performance by adapting 3D geometric complexity. This approach is essential since the geometric complexity of detailed scenes heavily impacts the rendering time. Model simplification, or level of detail (LOD), was among the earliest techniques used in conjunction with foveation. It is based on the observation that much of the complexity in a realistic 3D model is redundant when rendering the model from a given perspective since individual details may become too small to be perceived [137]. Standard adaptive rendering technique vary density based on factors such as *distance, size, velocity*, and *eccentricity* [149], as well as *semantics*, and *frame rate*[70]. Foveation techniques may be employed in isolation or in conjunction with these other approaches. Nowadays, gaze-tracked geometric simplifications are among the most widely used techniques to accelerate the rendering process[219]. Table 4 provides an overview of the different geometric simplification techniques used in foveated rendering. In the following, we will summarize the various subclasses of geometric simplification techniques and provide a general discussion of the state-of-the-art in this area.

6.1. Ray-based techniques

Ray-based techniques typically use *acceleration structures*, which achieve a rendering time that depends logarithmically on scene complexity. For this reason, geometric simplification is typically used only on very large scenes, and only a few studies explored ray-based methods for reducing geometric complexity in conjunction with foveation, especially in the case of dynamic gaze tracking. A representative example is given by the work of Weier et al. [241, 240], proposing a *ROI-based* geometric simplification model for large high-resolution display. The focus area is detected by tracing rays from the detected user position and intersecting the central viewing cone with the display. Since the display plane is seen at an angle, the authors model the focus area as an ellipse rather than a circle. Multi-resolution rendering is implemented by using the inner nodes of a *sparse voxel octree* data structure [122] as approximate representation, and the polygonal nodes of the original scene as a high-detail approximation. Due to the difficulty of rebuilding the sparse voxel octree on the fly, the system is tested only on static scenes. To individually decide when to stop traversing, a metric based on the distance of the ray to the central ellipse is used. Since hard transitions between levels are disturbing, the image at the periphery is blurred with a Gaussian filter with a fixed width. Similar user position-based LOD is also used in a rasterization pipeline by Scheel et al. [195] which is discussed in the next section. Other solutions, instead, produce continuous images by continuously varying the ray density and geometric LODs as a function of eccentricity. A representative example is given by the approach of Murphy and Duchowski [163]. In their approach, the scene geometry is sampled by ray casting, with a ray distribution conforming to the angular frequency dictated by a Contrast Sensitivity Function (CSF). This sampling generates an intermediate mesh, which is then further refined to preserve silhouette edges and rendered in place of the original geometry. One notable finding from this study is that the search time decreases with the foveated window size increment (up to 10° eccentricity).

6.2. Raster-based techniques

Raster-based techniques that adapt geometric complexity at each frame to meet performance constraints are the most classic approach for time-critical rendering [257]. Early approaches (e.g., [70, 149]), already used heuristic functions based on eccentricity with respect to a static gaze point (typically the screen center) to determine the level of detail. Use of the CSF for view-dependent polygonal simplification is also well established (e.g., [136, 184]). The acuity fall-off models used in these early works were later extended to dynamic gaze situations, in conjunction with eye trackers.

In an early geometric simplification model developed by Ohshima et al. [167], six different levels from the set of *hierarchical geometric models* are selected to be rendered according to the Euclidean distance from the ROI. In addition, this model exploits HVS subdividing the visual regions into *central, peripheral, kinetic*, and *fusion* zones. It is interesting to note that, since discrete LOD switch causes notable artifacts, the updating is postponed during saccade movements. While the method is designed for eye-tracking, the presented results were only for a head-tracking situation. Later approaches switched to continuous LODs to provide a much finer adaptation granularity and reduce LOD switching artifacts. Luebke et al. [137], in particular, used a multi-resolution mesh model supporting view-dependent-simplification to propose gaze-directed geometric simplification technique based on *contrast matching function* and *Kelly's temporal contrast sensitivity function* [108]. Results demonstrate good quality images with only one-third of the total number of polygons for benchmark scenes. However, in their implementation, temporal contrast sensitivity is not considered. Murphy et al. [162] also used a multi-resolution mesh representation to render objects in a gaze-contingent manner. This is achieved by recursively subdividing triangles that are larger than the local resolution provided by an acuity-based function depending on eccentricity with respect to the gaze point. This is the first study to use binocular eye tracking inside

Table 4. List of different geometric simplification techniques; similar techniques have been clustered together.

Algorithm used	References	Static	Dynamic		Pipeline	
			Eye-tracker	Gaze-tracker	Ray-based	Raster-based
Mesh simplification	Ohshima et al. [167]	○	○	*Head*	○	●
Textured mesh simplification	Luebke et al. [137]	○	●	○	○	●
Polygon simplification	Luebke and Hallen [136]	●	○	○	○	●
Texture simplification with 3D mipmap	Funkhouser and Sequin [70]	●	○	○	○	●
Level of detail (LOD)	Reddy [149, 184]	●	○	○	○	●
LOD	Murphy and Duchowski [162], Parkhurst and Niebur [170]	○	●	○	○	●
LOD	Scheel et al. [195]	○	○	*Optic*	○	●
LOD	Bektas et al. [27]	○	●	*Mouse*	○	●
LOD (holo)	Ju and Park [103]	○	○	*Mouse*	○	●
Adaptive tessellation	Papadopoulos and Kaufmann [168]	○	○	*Head*	○	●
Adaptive tessellation	Lindeberg [131], Zheng et al. [262]	○	●	○	○	●
Adaptive tessellation	Tiwary et al. [219]	○	○	*Mouse*	○	●
Curvature 3D simplification	Cheng [44]	○	●	○	○	●
Sparse voxel octree	Weier et al. [241, 240]	○	○	*Optic*	●	○
CSF-based ray mask	Murphy et al. [163]	○	●	*Head*	●	○

Head = head-tracker *Optic = Optical-tracker*

a head-mounted display. These general LOD-based approaches were later applied, with minimal variation, for a variety of different applications, including rendering models coming from 3D scanning [44] or large terrains [195]. In a visual search study using an eye tracker on a desktop display, Parkhurst and Niebur [170] rendered objects at the point of gaze in more detail than objects in the periphery. They found that, while search times increase with decreasing LODs beyond a critical threshold, the resulting increase in frame rate facilitates virtual interaction. Later studies found that contrast is a better predictor of the overall search performance and perceptibility than feature size, and, thus, variable resolution rendering is mostly beneficial if detail is added to low contrast regions first [234]. LOD rendering is also used in conjunction with non-conventional displays. For instance, Ju and Park [103] exploited levels of detail to speed-up the generation of *computer-generated holograms* for AR applications on a near-eye holographic display. The algorithm computes the angular spectrum of individual meshes, aggregates them in a hologram plane, and then Fourier transforms them to produce the complex wave field of the entire scene. LODs are used to adapt the density of meshes so that they are higher at the fovea. Adapting the mesh density through mesh adaptation improves over the prior point-based approach [238, 85, 84, 100] that simply adapts point density, leaving vacant areas between points.

One of the main limitations of early LOD techniques was the low granularity of LOD approaches and the limited performance of continuous LOD solutions, which made them difficult to apply in the very time-constrained setting of foveated rendering. Several of the later methods started to take into account the evolution of GPUs by amortizing LOD computation efforts on groups of primitives (e.g., surface patches), rather than computing the required level-of-detail at the single triangle or point level [47, 75, 76]. With this approach, CPU utilization was minimized, and applications could very quickly adapt the resolution even when dealing with massive scenes. This solution was adapted, e.g., for view-dependent rendering on a light-field display [31].

As an alternative to batching, several solutions have recently exploited *GPU tessellation* to achieve the fast adaptation time required by foveation applications. Lindeberg [131], for instance, introduced a *depth of field tessellation*, in which in conjunction with the reduction of tessellation levels our of the focus plane, there is an increase of blurring with eccentricity. Importantly, the user study shows that *pop-up artifacts* significantly decrease with the increase in blur level, suggesting that the technique can be used to hide the *pop-up effect*. An alternative solution, proposed by Tiwary et al. [219], instead, is to perform calculations of tessellation levels only during saccadic motions and to adapt the mesh only at fixations. Swafford et al. [210] propose a method in which imperceptible triangles are culled and then a tessellation shader parameterized with the acuity fall-off model is applied. A similar approach was also proposed by Zheng et al. [262]. Under multi-tiled LCDs, Papadopoulos and Kaufmann [168] present acuity-driven 2D gigapixel imagery visualization using a *GPU-tessellation* scheme for high-quality focus plus context lens and virtual texture rendering. The tessellation level of the context area of the image and of the lens is calculated differently, taking into account both the position of the viewer with respect to the screen and the deformation applied by the lens. The results indicate that using the high-quality focus plus context lens significantly reduces visual artifacts while accurately capturing the underlying lens function. Moreover, their parallel system saves up to 70% of the bandwidth and achieves frame rates of 7.5 fps, compared to less than 2 fps for naive pre-tessellation that does not take into account the user's gaze.

6.3. Discussion

All systems dealing with complex scenes to be rendered within stringent real-time constraints must integrate techniques for filtering out as efficiently as possible the data that is not contributing to a particular image. The goal is to have rendering complexity proportional to the bounded perceivable image size rather than to the potentially unbounded scene size. View-dependent geometric simplification has been one of the major building blocks of real-time systems in this particular context [257]. In the context of foveation, the general solutions are adapted to the particular conditions in which these techniques must operate.

First of all, several approaches include the definition of adaptive metrics that drive simplification refinement based on perceptual measures specific to foveation. Currently, no single approach has emerged as a de-facto standard, and techniques range from using just pre-determined simplification levels at the center or the periphery (e.g., [241]) to locally adapting sampling rates based on perceptual functions e.g., [137, 163, 168]). Many of the methods adapt these functions to display-specific situations.

Second, while typical adaptive rendering solutions slowly and smoothly vary tessellation as a function, e.g., of distance to the viewer, foveated solutions tend to be effective when simplification is applied in a much more aggressive way, with a sharp decrease in details outside of the focus area. The low level of detail in the periphery, however, is prone to introduce visible flickering artifacts. For these reasons, geometric simplification techniques are seldom used alone, but are often combined with a screen-space technique that blurs the low-detail areas (e.g., [241, 240, 131]).

Finally, knowledge of gaze provided by high-frequency and high-precision trackers can be exploited to schedule computations and adaptation during the saccade and/or fixation periods, with the purpose of reducing costs and improving visual fidelity (e.g., [167, 219]).

7. Shading simplification and chromatic degradation

While the previously discussed classes achieve optimization by reducing the number of rendered pixels or geometric primitives, the *third group* of techniques in our classification (Figure 5) achieves optimization by adaptively reducing the work or data required per pixel. We dedicate shader simplification (Sec. 7.1), and chromatic degradation (Sec. 7.2) under one single category, because the works pursued in these categories have the *common goal* of condensing the computation load of computing a photorealistic representation. However, while shader simplification reduces the computational load of color computation, chromatic degradation takes into account variable color sensitivity, e.g., to reduce bandwidth or complexity of tone mapping.

In the following subsections, we first present an analysis of recent literature on different shading simplification models (Sec. 7.1), and then investigate different techniques developed for chromatic degradation (Sec. 7.2).

7.1. Shading simplification

In advanced photorealistic rendering, as well as in illustrative rendering, computing the final color of each pixel may consume a significant proportion of computing resources, even for geometrically simple scenes. In recent years, several real-time graphics solutions have been employed for reducing rendering loads through the reduction of shader costs. A notable example is *Variable Rate Shading (VRS)*, introduced in *DirectX 12* graphics pipeline [205]. In foveated rendering, shader simplification optimizes the rendering time by using higher accuracy, but slower, methods in the focus area and simplified, but faster, ones in the periphery. The techniques include coarse shaders, multi-rate shaders, lighting, and occlusion simplification. In this section, we provide an analysis of the literature in this area.

7.1.1. Methods

In the context of shading simplification, there is not a sharp difference between ray-based and raster-based techniques, since most works use hybrid approaches. The most common configuration consists in ray-based shaders executing within a raster-based pipeline.

The fact that shaders can be used to naturally simulate general gaze-contingent stimuli was recognized early on. In particular, Duchowski and Coltekin [60] developed the first gaze-dependent fragment shader in which visual stimuli, such as color and luminance values were discarded in the periphery. This approach was designed, however, for foveation simulation, and not for optimization, and was used in a variety of applications. For instance, in their *space-variant visualization* framework, Bektas et al. [27] implement the degraded quality using pixel shader (GLSL language). This gaze-contingent display also can manage the level of detail (LOD) using a weighted Euclidean distance between any pixel and the gaze point in 2D space.

Later, shader techniques were also employed to reduce workload in addition to simulating foveation effects. Since shader simplification works well when the high-quality shader must do complex computations, the technique is often applied when using global illumination models, which must perform integration to aggregate realistic lighting information. Moreover, due to the inherent real-time adaptation features, these methods adapt well to dynamic gaze-tracking.

Table 5. List of different shading simplification techniques; similar techniques have been clustered together.

Algorithm used	References	Static	Dynamic		Pipeline	
			Eye-tracker	Gaze-tracker	Ray-based	Raster-based
Gaze-contingent occlusion	Mantiuk and Janus [144]	○	●	○	○	●
Screen space ambient occlusion	Mantiuk [143]	○	●	○	○	●
Coarse pixel shader	Vaidyanathan et al. [225], He et al. [82], Xiao et al. [248]	○	○	*Virt.*	○	●
Coarse pixel shader	Patney et al. [173, 172]	○	●	○	○	●
Multi-rate shader	Stengel et al. [203]	○	●	○	○	●
Pixel shader degradation	Duchowski et al. [60]	○	●	*Mouse*	○	●
Gaze-contingent pixel shader	Bektas et al. [27]	○	●	○	○	●

Virt. = virtual camera

For instance, global illumination with the *ambient occlusion shader model* improves photorealism through shadowing the ambient light of nearby objects. Mantiuk and Janus [144] propose a gaze-dependent hybrid model in which the ROIs are rendered with *ambient occlusion*, with a number of ambient occlusion sampling rays decreasing with eccentricity, and areas outside the ROI with local Phong shading. On the presented benchmarks, the method achieved a performance boost up to 276% in the best-case scenario, and on average 140.07% without negatively affecting user performance. The approach was later extended by the same authors to gaze-dependent *screen-space ambient occlusion (SSAO)* [143]. In the implementation, ROIs have 32 samples per pixel, while the sampling rate is gradually decreased with higher eccentricity according to the CSF.

Adjusting the number of samples has then been generalized to control variable shading rates (VRS) in a GPU pipeline. In their seminal study, Vaidyanathan et al. [225] introduced the first coarse pixel shader (CPS), derived from *multi-sample anti-aliasing (MSAA)* [5]. Generally, MSAA uses a fixed number of visible samples; however, the CPS allows predefined varied shading samples across the image. As a result, the number of shading computations on the shaded quads saved is about 50% than Guenter et al. [80]. Similarly, Patney et al. [173] apply variable-rate shading at different resolutions which enable coarse rendering after 30° eccentricity. In addition to shading reduction, one shader for each 4×4 pixel-block, blur mask, contrast enhancement, and temporal anti-aliasing (TAA) is used to discard peripheral visual artifacts. As an improvement, this approach decreases the shading rate by up to 70% in comparison to Guenter et al. [80]. Furthermore, Patney et al. [172] demonstrate a set of perceptual-based methods to enhance immersion experience and alleviate the computational burden of VR using 8× MSAA to ensure temporal stability in foveated rendering. He et al. [82] demonstrated that simple pipeline mechanisms present in programmable GPU hardware used in conjunction with adaptive shading techniques that select whether to use 2×2 coarse or fine fragments for shading can reduce the cost of shading during rendering by at least a factor of two in most benchmarks. More complex pipeline scheduling enables using even coarser fragments (up to 4×4 groups of pixels, reducing shading costs, on average, to more than three and sometimes up to a factor of five. Nowadays, VRS [205] is now a hardware-implemented solution available in graphics pipelines. For instance, the Turing architecture from *NVIDIA* combines VRS [205] with adaptive resolutions [29] to speed-up rendering. This approach can be exploited in foveated rendering by decreasing the shading rate in the periphery through perceptually guided measures [82].

The above decoupled sampling techniques, such as coarse pixel shading, is that they reduce costs by lowering the shading rate while resolving visibility at the full resolution, thereby preserving details along geometric edges. This is a major advantage with respect to several of the sparse visibility sampling methods of Sec. 5 or the geometric simplification techniques of Sec. 6. However, loss of texture details can produce visible blocking artifacts and temporal jittering in the periphery. For this reason, Xiao et al. [248] propose to combine coarse shading *temporal supersampling*, i.e., jittering frames and combining samples from multiple frames together. While not originally applied to foveation, this method is at the basis of several spatio-temporal techniques (Sec. 8). Stengel et al. [203] generalized the concept of multirate shading by incorporating shading rate adaptation in a flexible *GPU deferred rasterization*. In their approach, several properties of the sampling scene are accumulated in buffers during the geometry pass. These include, in addition to the usual depth, normal, and material information, also velocity and semantic information. A perceptual pass combines an acuity falloff function with several other hints, such as eye motion, texture adaptation, silhouette, eye adaptation to luminance, to produce a sampling probability map, from which a sparse sampling pattern is generated. The pattern is stored in the depth buffer, and early-depth is used to stop processing unselected fragments. The final images are produced by applying an inpainting process. This approach is very general and has been shown to decrease the number of shaded fragments by 50%-80% in comparison to the prior works (e.g., [225, 82, 80]).

7.1.2. Discussion

Shader simplification is an extremely effective technique to reduce the overall cost of rendering on high-resolution displays since the pixel shader is often the dominant factor. Modern shader simplification performs coarse rendering in the periphery with either stochastic sampling and inpainting [203], or reduced shading rate [82] followed by advanced filtering [173, 172]. The implementation of gaze-dependent shader optimization has been simplified with the introduction of CPS and VRS as common features in modern GPUs, such as *NVIDIA Turing* and *Intel Gen 11* architectures. Specialized solutions need; however, to be devised to aggressively apply CPS in a foveation setting. First, since CPS is unmatched with the visible samples, jittering and flickering are frequently generated in the overly simplified area at the periphery of foveated renderings. These dynamic artifacts are known to be visible and require the application of strong temporal-anti-aliasing methods. Second, the rendered scene has lower shading quality in the disoccluded regions, especially as it is more visible during fast motion or dynamic shading.

7.2. Chromatic degradation

Achromatic (luminance) spatial acuity in the HVS is known to be better than chromatic spatial acuity [161]. Video and image codecs have exploited this fact by separating signals into luma and chroma components and reducing the amount of color information in a signal in favor of luminance data [246]. Color sensitivity also rapidly decreases in the peripheral region like any other type of visual stimuli. It is thus possible to perform chromatic degradation in the non-focal areas without negatively affecting the perceptual quality of the images. This process can be exploited to, e.g., to perform gaze-dependent tone mapping or reduce the required bandwidth for the storage and transmission of images, especially high dynamic range ones.

This section comprises different techniques developed for chromatic degradation in the periphery. Table 6 lists several techniques used for chromatic degradation.

7.2.1. Methods

As for shading simplification, there is not a sharp difference between ray-based and raster-based techniques, since chromatic degradation happens at the level of color computation.

Several works in these areas are centered around user studies to find the tolerable color degradation in the periphery. Among other techniques, Zhang et al. [259] develop a peripheral color tolerance model based on the *CIE2000* color difference formula. In this technique, the individual chromatic discrimination models at parafovea and periphery are stored in a look-up table for future use. Duchowski et al. [59] develop color degradation maps by assigning each pixel's *gray value* to its corresponding contour value. Apart from the original resolution degradation model, Watson et al. [235] also use chromaticity degradation by applying grayscale in the periphery. Bektas et al. [27] apply modified *color degradation mask* developed by Duchowski et al. [59], and integrate it in a general gaze-dependent framework for testing user performance on visual analysis tasks.

In one of the earlier studies on chromatic degradation, Sakurai et al. [194] investigate *color zone map*, in which each zone has three primary colors, and unique hue components that correspond to *temporal, upper nasal*, and *lower directions* in the visual field. One most striking finding is that, with eccentricity, the hue changes and saturation of unique hue components decreases. Likewise, the *hue resolution* also can be defined by the total number of gray levels within each RGB channel. Correspondingly, Liu and Hua [132] design spatial CSF-based *chromatic foveation mask*, and hue resolution foveation metric. Interestingly, this method has been shown to save bandwidth over 65% in *image transmission*.

When dealing with colors, it is important to note that *tone mapping* has to be used used for reproducing high dynamic range (HDR) colors coming out of the rendering pipeline to the color gamut of the display. Knowledge of gaze information has been shown to be important to improve this process. As noted in Sec. 3.8, the HVS is always slowly adapting to a target luminance measured in a cone of approximately 1 degree around the gaze direction. The gaze is; however, not static, but follows saccadic motions. Mikami et al. [157] introduced a gaze-dependent approach based on a parameterization of Reinhard's photographic operator. They measure the local adaptation luminance by examining ROIs of $2°, 4°$, and $10°$ around the viewing angles, and take as the final adaptation luminance the logarithmic average from the original compression equation. Experimental results demonstrated, however, that the results are very scene-dependent [252].

Mantiuk and Markowski [146] generalized this concept by proposing a gaze-dependent *global tone mapping* for HDR images. In their approach, for every pixel in the input HDR image, which may be the output of a complex rendering process, a map of the background adaptation luminance is computed. This is done in a GPU shader that analyzes a one-degree area around each pixel and defines the local adaptation luminance to the most frequent quantized luminance value in that area. This work is done only when the rendered image changes. At each frame, the gaze direction is captured, filtered, and used to compute the temporary adaptation luminance, which combines the fetched background adaptation luminance with the previous temporary adaptation luminance using an exponential function. The model describes adaptation to light, e.g., when the observer moves his gaze from dark areas to bright areas of the display. This adaptation luminance is then used to compute the tone compression curve and compress the HDR image. The work was later extended to

Table 6. List of different chromatic degradation techniques; similar techniques have been clustered together.

Algorithm used	References	Static	Dynamic		Pipeline	
			Eye-tracker	Gaze-tracker	Ray-based	Raster-based
Adaptive tone mapping	Mikami et al. [157], Yamauchi et al. [252], Mauderer et al. [152], Mantiuk [146, 142]	○	●	○	○	●
Color zone mapping	Duchowski et al. [59]	○	●	○	○	●
Color zone mapping	Sakurai et al. [194]	●	○	○	○	●
Color tolerance model	Zhang et al. [259]	○	●	○	○	●
Gray scale increment	Watson et al. [235]	○	○	Mouse	○	●
Degradation color mask	Bektas et al. [27]	○	●	Mouse	○	●
Spatial chromatic mask	Liu and Hua [132]	●	○	○	○	●

videos [142]. In this latter work, to avoid the artifacts, *ocular Modulation Transfer Function* [52] in linear luminance, and two Gaussian pooling filters in the nonlinear domain have been applied. Similarly, in a user study, Mauderer et al. [152] gradually degrade color using *tone mapping* to see the color discrimination effect in the periphery. Although this method improves color discrimination, the low eye-tracking frequency may generate flickering effects.

7.2.2. Discussion

While early methods, and many current works, study color degradation in the context of psychophysical testing, more recent work has started to exploit it for optimization purposes. The first area of interest is bandwidth reduction (e.g., [132]), which takes into account that lossy compression models can use gaze-dependent color sensitivity information to optimally allocate bitrates across a viewed image. The second area of interest emerging is tone mapping which, in the most general case, must definitely be gaze-dependent. While research has mostly targeted the gaze-dependent presentation of HDR content (e.g., [146, 142, 152]), such information can also be exploited to avoid intensive computation by combining it with shader simplification (see Sec. 7.1).

8. Spatio-temporal deterioration

The final and *fourth group* of techniques in our classification (Figure 5) strives to improve performance by adapting the refresh rate of pixels across the image, eventually reusing information from previous frames for the less important pixels.

Spatio-temporal deterioration is a feature found in many real-time, multi-rate, and multipass rendering algorithms, as it strives to amortize rendering costs over multiple frames. In foveated rendering, these techniques need to be suitably updated, as they need to take into account the temporal sensitivity in the foveal region, in the periphery, or both.

8.1. Methods

Temporal coherence strives to reuse the intermediate or final information computed during the course of one frame to speed-up the rendering of the following frames. As such, it complements the previously seen approaches, that focus on improving the performance of individual rendering tasks, eventually by lowering the accuracy at which one frame is computed. This general approach dates from the early days of graphics [208], and has led to a wide variety of approaches [196].

Foveated rendering has also used spatio-temporal deterioration approaches since its early days as a component of many frameworks. Dorr et al. [56] were among the first to present a gaze-contingent system capable of modulating the spatio-temporal contents of a high-resolution realtime video, but adapting the spatial multiresolution pyramid of previous approaches [72, 174] to a temporal pyramid. Moreover, several early peripheral pixel reduction methods (e.g., [80, 63]) applied a combination of motion-compensated temporal reprojection [79] and temporal jitter on a spatial sampling grid [49] to decrease frame times by recomputing a smaller number of pixel per frame in the periphery (Sec. 5.4). Since then, a wide variety of foveated spatio-temporal solutions were integrated in both ray-casting and rasterization pipelines.

Several approaches adapt classic optimizations, such as amortized supersampling [254, 248] an reprojection caches [165]. Weier et al. [243] presented a foveated real-time ray tracer combined foveated rendering based on dynamic eye tracking with reprojection rendering using previous frames to drastically reduce the number of new image samples per frame. A smooth image is then generated by combining these sparse samples with data coming from previous frames. First, a coarse depth mesh is reconstructed from the previous frame samples, and a coarse image is rendered from the current frame perspective. Then the parts of the image that are considered not valid due to occlusions/disocclusions/missing data or poor reprojections are identified. This is done by detecting if there is a depth or luminance difference between a current frame's pixel and its direct neighborhood in the reprojected image that is larger than a user-defined threshold or if the pixel

Table 7. List of different spatio-temporal techniques; similar techniques have been clustered together.

Algorithm used	References	Static	Dynamic		Pipeline	
			Eye-tracker	Gaze-tracker	Ray-based	Raster-based
Temporal raytracing	Weier et al. [243]	○	●	○	●	○
Temporal pyramid	Dorr et al. [56]	○	●	○	○	●
Time-warped rendering	Linus et al. [65]	○	●	○	○	●
Spatio-temporal filtering	Jiang et al. [97]	○	●	○	○	●
Temporal supersampling	Xiao et al. [248]	○	●	○	○	●

is on a silhouette edge. Finally, the high-resolution image is generated, reusing reprojected pixels from the previous frame whenever possible, and recomputing invalid pixels by ray-tracing. Reflections and refractions are reasonably well handled if present in small areas of the image, since those pixels are likely to be recomputed. Moving lights, however, tend to drastically degrade performance.

Franke et al. [65] used similar approaches in a rasterization pipeline. Since in rasterization redrawing single pixels cannot be done efficiently, their focus is to devise approaches to reduce expensive redrawing operations without visible impact on image quality. In their approach, the last frame's color and world position images are reprojected into the current frame and hole-filled using a push-pull filter [203]. A confidence map is then derived by combining an *eccentricity confidence factor*, based on the falloff in the eye's visual acuity with two factors that measure the confidence in hole-filling result. The first factor is inversely proportional to contrast, while the second is inversely proportional to the hole size. Moving objects are handled by lowering confidence of pixels where object motion is detected. All pixels whose confidence is below a given threshold are then redrawn. This is done by redrawing the scene, culling out objects that are totally covered by high-confidence pixels. Before displaying the final image, a TAA and motion smoothing pass is applied. The method proves very efficient, but is less capable to handle transparency and reflection than the fine-grained raytracing approach [243], while still being incapable to efficiently support moving lights.

8.2. Discussion

One of the main problems in adopting temporal degradation methods is that, unlike the spatial resolution as a function of eccentricity, the peripheral temporal characteristics of the HVS are still not totally understood [56]. This makes it difficult to have reliable models that predict the effect of spatio-temporal degradation. Recently, Krajanciche et al. [120] proposed the first experimentally derived comprehensive model for spatio-temporal aspects over the retina under conditions close to VR applications. It is interesting to note that temporal sensitivity has been observed to peak in the periphery, somewhere between $20° - 50°$ eccentricity [224, 120]. This means that foveated rendering solutions cannot limit themselves to just focus on providing high-quality rendering for the fovea, spending as little resources as possible in the periphery, but should also combat peripheral flickering. While those effects can be significantly amortized by spatiotemporal filtering [80, 63, 248, 97], these solutions are only partial, as they tend to overly reduce local contrast. Loss of contrast in a large area of the periphery region can result in tunnel vision artifacts [39]. For this reason, other authors have tried, with variable success, to produce flicker-control schemes that strive to preserve contrast [173, 97]. An important consideration to make is that the sensitivity to temporal artifacts also depends on fixation types. Weier et al. [243], for instance, noted that fewer visual artifacts were noticed when users concentrated their attention on a moving target, a fact that could be exploited in future work. Further considerations are presented in Sec. 10.1.

9. Applications

Foveated rendering may be viewed as a general optimization technique, which could be applied to any use case in which interactive images are presented to viewers. Nonetheless, in the past years, foveation has been applied more extensively in a few selected areas that we have broadly classified into visualization (Sec. 9.1), compression (Sec. 9.2), and transmission (Sec. 9.3). Compression and transmission are included here as they offer enabling technology for remote rendering and collaboration, and, for maximum efficiency, end-to-end systems require a careful integration of all components. Table 8 distributes the surveyed literature among these selected areas.

9.1. Foveated visualization

In this application class, we broadly classify all situations in which the main application of foveation is to visualize data, either to improve application performance or to display some effects to emulate particular viewing conditions.

9.1.1. Immersive visualization

According to Cuervo et al. [51], three parameters are essential for a truly immersive virtual experience: *quality, responsiveness*, and *mobility*. The *quality* guarantees natural and real-world visual experience, *responsiveness* represents rapid visual feedback to motion, and *mobility* allows moving untethered in physical space. Park et al.

Table 8. Different application domains of foveated rendering, most of the research engage in rendering and visualization. Compression and transmission are included as they offer enabling technology for remote rendering and collaboration, and, for maximum efficiency, end-to-end systems require a careful integration of all components.

Foveated visualization	Application in visualization (Sec. 9.1)	Foveated compression (Sec. 9.2)	Foveated transmission (Sec. 9.3)
Adaptive resolution (Sec. 5)	[127, 235, 174, 24, 23, 227, 77, 176, 3, 34, 78, 199, 260, 80, 63, 69, 201, 177, 27, 28, 209, 179, 180, 243, 85, 139, 123, 115, 116, 242, 156, 213, 100, 84, 226, 148, 214, 103, 169, 238, 68, 154, 36, 223, 110, 200, 117, 155, 247, 10, 256, 140, 253, 18, 175]	[72, 42, 198, 90, 106, 255]	[72, 258, 2, 158, 113, 138, 51, 92, 93, 189, 109, 156, 154, 169, 67, 211, 155, 94, 128]
Geometric simplification (Sec. 6)	[70, 167, 149, 137, 137, 136, 162, 184, 44, 170, 236, 163, 168, 241, 240, 195, 131, 262, 219]		[168, 195]
Shading simplification and chromatic degradation (Sec. 7)	[164, 60, 144, 225, 82, 27, 192, 173, 172, 143, 248] [235, 194, 59, 145, 69, 27, 259, 152, 142]	[132]	[203]
Spatio-temporal deterioration (Sec. 8)	[80, 63, 56, 243, 248, 97, 65]		

[169] also suggest that a display requires *high resolution without screen door effects, wide FOV, high frame rate without motion artifact*, and *minimum tolerable latency* for an immersive experience. Similarly, Fujita and Harada [69] report *fast, low-latency, smooth*, and *realistic rendering methods* are crucial for immersion. Weier et al. [243] support this statement by exploring the necessity of *high frame rate*, and *low latency*.

The higher demand on pixel density along with the stereo display increases the complexity of the real-time rendering process, making foveated rendering very appealing. With the emergence of robust eye-trackers that allow individual vision, immersive VR has now been considered the main application domain of foveated rendering. Seminal foveated rendering research for immersive experience are based on adaptive sampling [80, 243, 69, 156], coarse pixel shading [225, 82, 173, 172], rolling rasterization [68], and contrast aware foveation [223]. Due to peripheral degradation, immersion is not free from flickering, and a strong anti-aliasing algorithm is required. There is another downside of conventional VR displays. Because of the flat surface, the *vergence and accommodation conflict* stops the foveated window from acquiring accurate depth information. However, the modern near-eye displays, e.g., holographic, varifocal, and light field can overcome this drawback, which will increase the level of immersion, but with higher computation. In a recent review on near-eye holographic display, Chang et al. [45] concisely explore the potentiality of foveated rendering in holographic displays. According to the authors, foveated rendering is possible either with multiple display panels or on rendering technique. The potential rendering approaches can be point cloud [84, 100], polygon mesh [103] and multi-plane models [139, 41, 123]. Besides, Chang et al. [41] recommend that the first two approaches rely on complicated geometry and computer graphics processing. Nonetheless, the *multi-plane model* is much simpler and more efficient, in which the 3D scene is rendered as multiple planar 2D images.

9.1.2. Volumetric visualization

Volumetric data visualization has become more common nowadays due to the advances in 3D data acquisition and complex simulations on modern displays with an interactive framerate. Due to the enormous complexity of semi-transparent volume rendering, which requires the computation of integrals per pixel, maintaining interactive performance is very hard, and much research has focused on volume-specific optimization techniques [19, 32]. In this context, foveation promises to be extremely effective, as it can drastically reduce both the number of pixels for which to compute these integrals and the quality at which they need to be computed. For this reason, many applications have been studied. Among the various outstanding foveated volumetric rendering methods it is important to mention applications to importance-driven medical data visualization [227], arbitrary geometric object visualization [163], large scale geometric dataset interaction [136], general volume data visualization [36], depth peeling-based data visualization [260], and large scale scientific data visualization [10]. Foveated volumetric approaches have also been introduced over 15 years ago in the context of remote visualization (e.g., [258, 176]).

9.1.3. Large-scale visualization

Many important application domains, including 3D scanning, computer-aided design, and numerical simulation, require the interactive inspection of extremely massive models. Despite the continuing and rapid improvement in GPU hardware performance, the interactive rendering of these models using brute force techniques continues largely overloading state-of-the-art hardware platforms. For this reason, researchers have devised a variety of adaptive techniques for rendering approximate representations, filtering out as efficiently as possible the data that is not contributing to a particular image [257]. Foveation promises to be extremely effective in this context. For this reason, foveation was used very early on for a variety of massive-model rendering use cases in a variety of configurations. These include foveated terrain rendering on very large high-resolution displays [184, 195], visualization of voxel data on tiled displays [241, 240, 192], focus-and-context visualization and large image data visualization on multi-projector systems [3, 27, 26], projection display of cultural heritage artifacts [77, 78], as well as information visualization on large high-resolution displays [13] visualize large scale information.

9.1.4. Vision defection mapping

Nowadays, a large population suffers from vision defects like myopia, hyperopia, glaucoma, presbyopia, and astigmatism. Therefore, considering the space-variant vision characteristics, an accurate simulation of an individual's visual field can educate students, patients, and family members about the perceptual defects. Foveated rendering methods are the basic enabling technology for this application use-case. Perry and Geisler [174, 73] design a multi-resolution pyramid based vision simulation framework that can visualize the resolution map of a *glaucoma* patient. In the same way, Labhishetty et al. [121] investigate accommodation conflict on *myopia* patients. Interestingly, this study suggests that, unlike fovea and perifovea, the *parafovea to higher eccentricity* is affected by *myopia*. Since rendering the resolution of non-foveal simulations can affect user accommodation, the authors suggest considering foveated rendering algorithms for such medical conditions. Fridman et al. [66] simulate observer vision with gaze point. Likewise, Deza et al. [55] visualize real-time *metameric image* using foveation. Correspondingly, Barsky [21] demonstrate computer-generated images that incorporate the characteristics of an individual's entire optical system based on the optical *wavefront aberrometry* measured using a *Shack-Hartmann aberrometer*. In fact, this study can also be used for efficient interface design, usability, safety, and behavioral evaluation. Recently, Wu and Kim [247] develop an AR display in which a free-form image combiner allows embedding *prescribed lens* to provide vision-corrected augmented object with an optical see-through display.

9.1.5. Preview systems

Several algorithms are too slow to fully produce full-quality, large FOV images at an interactive rate. Progressive rendering has been employed for decades in this situation to quickly provide coarse approximations to the viewer in a very short time [74]. Foveated rendering can be very beneficial in this area, by concentrating image improvements on areas that are currently viewed by the user. Koskela et al. [115] use the first approach developing a gaze-directed guided preview with the *quadratic denominator visual acuity model*. In this algorithm, more rays are generated around the ROIs using unidirectional path tracing. Unsurprisingly, this foveated preview system performs ten times faster than the conventional uniform sampling over the whole 360° image area, with little degradation with respect to uniform refinement [116].

9.2. Foveated compression

In several situations, rendering applications must work in a distributed setting. In that case, reducing the bandwidth of transmitted rendering images is particularly important. Foveation has been demonstrated to improve compression by considering gaze in bit allocation methods. Back in 1998, Geisler et al. [72] were among the first to advocate foveation for lossy video compression. Among the few representatives of foveated compression techniques, Sheikh et al. [198] developed gaze-contingent low-pass filtering on standard video compression algorithms (H.263 and MPEG4). Likewise, Wilson and Jeffrey [72] designed a multi-resolution *image compression* for low-bandwidth communication. It has, however, been noted that considerable savings are obtained only by aggressively reducing the quality outside the ROI, which can cause noticeable artifacts in the periphery. More conservative applications resolve these problems but provide only modest savings with respect to non-foveated compression [172]. Nonetheless, Frieß et al. [67] have successfully used this paradigm by proposing different parameterized macroblocks based on an H.264 encoder, considering an acuity fall-off. In their approach, the hardware encoder and foveated encoders have been merged to enable high-quality screen capture between two displays over a standard Ethernet connection (100-400 Mbps) for supporting remote collaborative visualization on large high-resolution displays with more than 44 megapixels. Recently, it has been demonstrated that the quality limitation problems of standard transform encoders may be overcome by deep learning approaches, in which deep networks are trained to reconstruct peripheral areas from very sparse samples [106]. These results are extremely promising, especially in the context of emerging 360° video formats [255]. As hardware-accelerated real-time video codecs integrated with GPUs have now become an essential enabling technology for many real-time graphics applications running over the network, e.g., cloud gaming [197], it is expected that future foveated codecs would be of even larger importance in VR settings [94]. For maximum benefits, it is important to integrate compression solutions with renderers, so as to avoid spending time on pixels on which few bits will be allocated.

9.3. Foveated transmission

Foveated transmission attempts to conserve bandwidth by sending only detailed information in the ROIs and lowering it to the periphery. Video transmission consumes most of the bandwidth over the internet. For instance, in

2019, 72% of the total mobile data traffic has been used for video transmission [138]. For this reason, much of the work concerning foveation has concentrated on improving general video transport for streaming services [193]. In this context, notable video transmission methods designed to concentrated effort on the fovea and reduce it in the periphery are gaze-dependent multimedia transmission [138], log-polar transformation [156, 154, 155], log-rectilinear [128] transformation, gaze crop filter [189] and likelihood-based foveation [51]. A notable result has been presented by Kim et al. [109], who developed the first foveated video player based on *MPEG Dynamic Adaptive Streaming (DASH)* over HTTP and *Spatial Relationship Description* for high definition 360° video streaming. In this approach, the scene is first subdivided into different regions. After the decoding of the regions, bit-stream stitching and 3D texture mapping are applied. Finally, a multi-resolution rendering is used where the center viewport is rendered with full resolution, four sides with 1/2, and corners with 1/4 of the resolution. However, while the authors claim that frame rates can be improved by 10%-15%, there is no solid evidence to back up this assertion. Likewise, Rondon et al. [189] designed a client-server system based on *bilayer resolution* and MPEG-DASH principle that streams only high-resolution 360° videos over ROIs. In the implementation, generating a one-second-long segment of 30 frames, server delay is approximately 700 ms per segment, or ca. 23 ms per frame, closer to tolerable latency.

Since minimizing *end-to-end* latency and maximizing refresh frequency and image quality is essential for VR. Thus, foveated transmission is also becoming a basic block for remote and collaborative interactive applications, which require a very close cooperation between rendering and transmission components.

In *remote visualization*, there are two techniques possible: *render local* and *render remote* [30]. For the *first* approach, the entire data volume is sent to the client device for rendering which requires high bandwidth. Aside from bandwidth, the requisite computation power is mostly unavailable for many low-end devices (e.g., tablets, smartphones). The *second* technique where data can be rendered at the server and then sent to the low-end devices, is more robust in that case. With an additional gaze-tracker, remote rendering has opened a whole new application domain like *foveated cloud gaming*, that allows playing high-end games on low-end devices, where low system latency is crucial [158, 92, 93, 46]. Illahi et al. [94] recently demonstrated that using a parameterized Foveated Video Encoding for real-time interaction in cloud gaming reduced bandwidth up to 10%.

Through foveated rendering, large-scale collaborative data visualization in a remote server has been demonstrated over standard bandwidth [258, 67]. In this context, Papadopoulos and Kaufman [168] designed a 1.5 gigapixels immersive display that can visualize both 360° videos and a large scientific data set over an internet browser. In addition to transmission, Syawaludin et al. [211] develop a dual-camera setup for 360° video-based remote interaction. Among the two cameras, one is a *pan-tilt-zoom camera*, and another is an omnidirectional camera but with the same frame rate.

Foveation has also been applied for the interactive capture and transmission of volumetric videos taking into account special 3D display characteristics. In particular, the data processing and transmission load for *light field displays* require an exceedingly large bandwidth and computation resources. Adhikarla et al. [2] developed the first light field data compression algorithm for a telepresence application on a large-scale light field display. The method takes into account display geometry and viewer positioning for discarding unused parts of the images from a camera array in the acquisition site before transmission. For a 19 second footage, this compression used only 20% of the whole data stream without introducing temporal or spatial artifacts. The approach was later extended to perform retargeting to different light field displays through adaptive depth range compression [113]. As the method generates a depth map, it can be used to combine both synthetic data and captured video. Thumuluri and Sharma [217] later designed a light field data reconstruction technique that claims faster data transmission.

10. Discussion

Foveated rendering has witnessed substantial progress in the past decades, growing from early methods aimed mainly at psycho-physical testings or proof-of-concept renderers to a variety of solutions for optimizing the rendering process in a variety of very demanding settings. Moreover, many of the proposed technical solutions have been used in a wide variety of realistic applications.

Our survey has provided an integrative view into this wide array of methods, highlighting the strengths and limitations that currently exist in the field. On the basis of this analysis, we provide a view of open problems and current and future works.

10.1. Improving current foveated rendering techniques

Foveated rendering is a potentially a very effective approach to jointly optimize rendering fidelity, frame rate, compression, transmission, and power consumption by adaptively varying peripheral image quality. Many techniques have been proposed in the past, that we have classified into four main peripheral degradation categories (Sec. 5-8). While the survey clearly demonstrates large advances in each of these categories, various bottlenecks still exist, leaving large space for further research. This is due, in particular, to the fact that, in most situations, foveation provides significant benefits especially when the focus area is maintained as small as possible, and very aggressive simplifications are applied. Under these conditions, even the best available techniques are prone to introduce visible artifacts on non-trivial scenes.

Spatial artifacts due to insufficient density of rendered images are an obvious outcome of foveated rendering approaches, especially on several display kinds that strive to offer a wide FOV coverage. For instance, maintaining high pixel density is crucial for minimizing stochastic visual artifacts, especially for near-eye displays. For instance, it is now common to combine two displays, one with high pixel density and another with low pixel density a near-eye AR display that reduces both pixelation and screen door effect (e.g., [213]). However, under even moderate degrees of foveation, the low-pixel density displays in the periphery often suffer from *staircase artifacts* and *motion aliasing (flickering)*. In addition, many other spatial artifacts may arise from the individual techniques employed to reduce rendering complexity. For instance, spatial edges are often visible in between layers created by the foveation [72], pupil swim effects may be the result of techniques that decompose a 3D scene into 2D layers heloing [123] and haloing and occlusion/disocclusion problems may arise from adaptive sampling approaches [144]. Moreover, *temporal artifacts* remain among the most common problems arising in foveated rendering, independently from the peripheral degradation technique employed. This is because the HVS is particularly vulnerable to temporal instability. In fact, peripheral vision is particularly sensitive to contrast changes and movements as the rods are highly concentrated at the periphery (maximum density at about 17° of the viewing direction) [65]. Peripheral vision, like the fovea, is also essential for intuitively perceiving the surroundings and reacting to changes and movement. Moreover, when motion starts, for instance: head rotation, eye movement, or animation, any visible aliasing effects (e.g., a lower spatial resolution) can create perceptible temporal artifacts, a.k.a., flickering. Surprisingly, the peripheral vision is more flicker sensitivity than even stereoscopic depth perception [224, 8]. For this reason, flickering is possibly the most common visual artifact in foveated rendering that often breaks the seamless visual experience. A wide number of solutions have been proposed to combat these problems, including blur mapping [22, 106, 150], depth of field filters [242, 91], temporal smoothing filters [36], phase-aligned rendering [25, 222], as well as display designs that strive to eliminate illumination variations [68]. All these solutions, though effective, have their pros and cons. For instance, blur also diminishes the local contrast [22, 106, 150]. This contrast reduction may lead to further visual artifacts, such as screen-door effect, pop-up effect, spatial-edge artifacts, temporal aliasing (flickering), and pupil swim effect [172, 97]. Moreover, temporal filters are also prone to contrast reduction and not easy to combine with many of the adaptive rendering techniques [118, 36, 155, 97].

10.2. Exploiting machine learning for foveated rendering

Efficient foveation techniques must quickly determine the gaze point with the minimum latency and exploit it to rapidly present a suitable approximation. This requires not only advances in tracking and display hardware but also advances in models for predicting eye motion to reduce latency and for determining image approximations that provide the best quality within the available resource budget. While many first-principle solutions have been proposed with various degrees of success (see Sec. 5-8)), one of the emerging research directions is to learn these models from examples (see Table 9). Replacing or augmenting trackers with an accurate gaze prediction model can reduce both computing complexity and latency (see Section 3.5). Research in this area is only starting. For instance, Lemley et al. [125] attempted to predict eye-motion through CNN architectures trained on the *PoG dataset* [153], and later improved the approach using use an *appearance-based CNN model* [126] on MPII-Gaze dataset [261]. Arabadzhiyska et al. [14] present another end-to-end *amplitude-based user-specific saccade prediction model*; however, two user experiments prove that the user-specific model predicts better saccade landing prediction than the general observer model, highlighting the difficulty of devising general approaches. Similarly, Mohammed and Staadt [159] model gaze-movements on a 4×6 multi-LCD high-resolution display with two reinforcement learning models, training and testing them on the Microsoft Salient Object dataset [133], and York University Eye Fixation dataset [35]. These approaches show the interests of the approach, but also highlight that current solutions are not robust to user-specific, and display-specific.

Learning techniques are also starting to deliver results also in the area of rendering. In particular, Fridman et al. [66] developed the first *Foveated Generative Network* and an online tool, *SideEye* for peripheral vision simulation, and Deza and Jonnalagadda [55] proposed another deep learning-based framework to construct visual metamers *NeuroFovea* in real-time. Moreover, Kaplanyan et al. [106] explored the usage of generative adversarial neural networks to reconstruct a plausible peripheral video from a small fraction of pixels provided every frame. The method, fast enough to drive gaze-contingent head-mounted displays in real-time on modern hardware, is shown capable to produce visual experiences with no noticeable quality degradation using only 10% of the pixels. Likewise, Thumuluri and Sharma [217] designed generative adversarial neural networks for light-field reconstruction, also using 10 times less light field data than the existing state-of-the-art work.

These early results show that the use of machine learning to improve foveated rendering is a promising but still not a fully explored research domain. Matthews et al. [150] suggest that, in general, multi-rate shading is not restricted to foveation and can be robustly implemented using a neural network model. However, among the existing research challenges is the relative shortfall of training databases, which are not easy to synthesize.

10.3. Supporting multiple users

Foveated rendering is a view-dependent rendering optimization technique, and foveated algorithms are typically

Table 9. Notable foveated machine learning approaches, relevant platforms, applications, and used database.

Reference	Platform	Applications	Technique	Database
[125]	AR/VR	Gaze prediction	Generative adversarial networks	PoG dataset [153]
[126]	AR/VR	Gaze prediction	CNN	MPII-Gaze dataset [261]
[159]	LHRD	User gaze model	MaxEntropyIRL, FIRL	MS Salient Object [133], York University Eye Fixation dataset [35]
[14]	Desktop	Saccade landing prediction	Parameterize amplitude model	In-house dataset
[106]	VR/Desktop	Video reconstruction	Generative adversarial-NN	YouTube-8M [1]
[217]	Light field display	Foveated reconstruction and view synthesis	Convolutional Neural Network	DeepFocus [249]
[4]	Desktop	Object detection	HOG feature, latent-SVM-like framework	PASCAL VOC 2007 [62]
[66]	Desktop	Peripheral vision simulator	generative-NN	Places dataset [263]
[55]	Desktop	Visual metamers simulation	Deep learning	In-house dataset

designed for single-view only. The near-eye and head-mounted displays are the most convenient for this intent. However, in several situations, multiple users can simultaneously watch a display, and single-user techniques are not directly applicable.

Regular small-sized displays makes it very difficult to take advantage of multiview foveation, since, in case of multiple users, much of the area of the display would be in focus. Even for large high-resolution displays, viewers are most of the time confined to the presenter's vision [195, 241, 240], and per-user foveation is still rare [65]. The increase in size and resolution of display surfaces, often combined with touch interfaces, and the need for remote and co-located collaboration makes multi-user foveation a very appealing alternative [67], and can be identified as a very interesting area for future research.

Non-conventional displays, which typically require much effort per pixel, are also offering important research opportunities. For instance, a light field display allows multiple users to watch a single scene from different perspectives, and, as noted by Spjuit et al. [102], efficient multi-user foveation is essential to avoid the computation of the very large number of rays not directed towards a viewer. Developing scalable and efficient techniques in these cases requires considerable research and engineering efforts, combining precise multi-user tracking with scalable, and often display-specific, low-latency parallel rendering methods taking into account foveation.

10.4. Evaluating the visual quality of foveated rendering

The advancement in foveation technology cannot be disjoint from advancements in methods for evaluating the visual quality of results. With foveated rendering, the graphics quality should be persistent and acceptable regardless of application specifications. While several efforts have been targeting evaluation, no consistent and standard evaluation method yet for assessing the foveated rendering quality, both subjectively or objectively.

Subjective evaluation is, in principle, very appealing, since it directly considers humans as the end-user of a graphics output [80, 173, 210]. However, it is also framework-based, scene-dependent, and observer-biased. Moreover, it is time- and resource-consuming, since the resulting scores need to be calculated from a decent amount of observers over multiple viewing sessions in which the observers confirm the foveated rendering is imperceptible than perceptible. A few authors have also suggested other qualitative measures than the pure ability to perceive or not variations, such as *efficiency* and *consistency* [88]. The *efficiency* of an experiment defines how quickly the perceptual ratio will converge with higher performance and lower experiment costs, such as shorter assessment time or fewer judgments. *Consistency*, on the other hand, seeks to assess the firmness of individual Quality of Experience (QoE) ratings. Only a few studies allow wearing eyeglasses during the evaluation [9]. There are also several testing approaches and statistical models used in the literature to evaluate qualitative result, such as 2AFC [223, 184, 65], MOS [138, 238, 219, 144, 143], ANOVA [27, 243, 152, 59, 9, 182, 235, 163], T-test [207], pairwise [163], and chi-square. Few other studies, such as [163, 243, 111] use multiple statistical models to validate their algorithms.

Objective evaluation based on quantitative measurements is often preferred by researchers because, the incorporation of models that predict outcomes for humans, leads to simpler ways to use the outcomes of the evaluation to drive adaptive methods. However, due to space-variant nature, the traditional *perception-based graphics quality matrices* [130] is debilitated in foveated rendering. A few research use conventional graphics quality metrics [40], e.g., SSIM [68, 169], DSSIM [154], PSNR [225, 169], but measure the foveal and peripheral graphics quality separately. Others, attempt to consider foveation-specific measures, for instance, the foveated wavelet image quality metric [230], that considers the spatial variance of CSF, local visual cut-off frequency, the Foveal Signal to Noise Ratio (FSNR), and Foveal Weight Signal to Noise Ratio (FWSNR), that consider the distortion visibility decrement in the periphery [124], and the Foveated Point Signal to Noise (FPSN) and Foveated Image Quality (FIQ) metrics for holographic displays [123].

Other authors have also proposed to adapt *full-reference* image quality metrics to foveated rendering. For instance, Tsai and Liu [220] sub-divides the scene into different window sizes, measures window scores using traditional and pool the scores together for an overall performance report. Other authors extend the the acuity fall-off model to compute foveated variations on standard scores, such as Foveated Mean Squared Error (FMSE) [188], Foveation Adaptive Root Mean Squared Error (FARMSE) [228], or the FLIP perceptual metric [12]. Noteworthy, such *full-reference* graphics quality evaluation is impractical due to the relative lack of reference in the graphics rendering process. Recently Mantiuk et al. [147] proposed a full-reference visual quality difference metric, *FovVideoVDP*. The metric can predict visual differences for different types of distortions: blur, JPEG compression, flicker, and Gaussian additive noise at different eccentricity levels, tested on rendering dataset, *FOVDOTS*. This metric is more efficient for higher FOV displays, such as AR/VR displays. However, color, glare, inter-channel masking, and eye motion were not included in the model, which requires further analysis.

Chen et al. [98] created the first compressed 360° video database, *LIVE-FRL* that can be used for foveated image and video quality assessment. This database consists of 190 videos with 8K quality, including 10 reference videos and 180 distorted or foveated videos which are also generated from the reference videos. Moreover, Jin et al. [99] published a study on both subjective and objective quality assessment of VR video compression, along with a 2D and stereo 3D video database. The complexity of foveated rendering quality evaluation and the high sensitivity to display and tracking characteristics makes it a very active research direction [210].

10.5. Studying the effects of foveation artifacts on user performance

While, ideally, the goal of foveation is to produce images indistinguishable from non-foveated ones, in practice some artifacts may appear in the rendered images. These artifacts may result from imperfections in tracking or displays, delays in various stages of the pipeline, or approximations in rendering methods or guiding metrics. Moreover, even in the case in which imperceptible images could be generated, it is often useful for applications to have the opportunity to trade image quality with speed, to come for massive/complex models or vary spatiotemporal realism depending on tasks.

A large set of studies in cognitive psychology have identified two interrelated classes of visual processing, referred to as *preattentive* and *attentive* vision, respectively [87]. In this model, preattentive vision scans large areas noting features that represent changes in pattern or motion. These features include color, size, luminance, motion, patterns, shape, orientation, curvature but not closure, gaps, or terminators. Attentive visual processes refer, instead, to processes required to recognize details about objects and relationships in scenes. In an early study, Watson et al. [232] suggest that, due to these human visual system characteristics, dynamic LOD control has to be content and task-dependent. As a result, during operations such as visual search, the observer necessitates more global visual information, leading to less foveation. Multiple studies have, thus, studied various forms of degradation during visual search tasks, to find how imperfect foveated displays affect visual performance. Other authors have concentrated their efforts on finding good central area sizes in which models have to be rendered at full resolution for gaze-contingent displays. Results vary from as large as around 10° [163] to less than 2°[26, 134] depending on the display, frequency of update, and image content. The same experiments performed on a desktop monitor and a near-eye VR display also show a wide variation (e.g., from 2°-5° for the monitor to 30° for the near-eye VR display). As noted very early by Watson et al. [233], however, viewers are more sensitive to how degraded are LODs in the periphery than the reduction of the central area.

While much of the research has concentrated on the degradation of resolution and geometric detail, chromatic sensitivity has also been shown to have important effects (see Sec. 7.2). Due to the complex inter-relations between physiological and psychophysical factors, it has been shown that color sensitivity is task-dependent, and that, for search tasks, color precision cannot be reduced in the same way as visual acuity [59]. For instance, when the spatial detail is lowered by 50% after a 5° viewing angle, the chromatic reduction should not be dropped before 20°, otherwise, deterioration may become visible. This task dependence is also emphasized by the differences in outcomes of several user studies. Hansen et al. [81] recommend that the color sensation becomes more *dichromatic* at about 25°-30°, due to the lack of L and M cones, and becomes absent at eccentricity after 40° for *weak stimuli*. However, Ayma et al. [17] conduct color zone mapping with two user experiments in which the results prove that color perception is even better above 20° eccentricity; but, from the mid-periphery ($\approx 40°$), the red-green hue appears to be less chromatic than yellow-blue due to the *post-receptoral cortical process*. Similarly, Buck et al. [37] suggest that the fovea-like color vision still exists out to at least 45° eccentricity. Besides the eccentricity, the stimulus size is also a critical and crucial parameter for color perception. Noorlander et al. [166] analyze that under specific spatial and temporal conditions, such as a large target size and low temporal frequency (1 Hz), different hues can be perceived at the eccentricity of up to 90°. However, color perception is not constant across the life span. Webster et al. [237] prove that the color degradation even is visible after near periphery (8°) because of aging.

The high variability in reported results and the dependence on display, content, and degradation techniques indicates that considerable research is still required to find good ways to aggressively degrade quality in the center and periphery without impacting search performance.

11. Conclusion

This survey has provided an integrative view of the domain of foveated rendering, focused mainly on the techniques that have been employed to perform the optimization. Our first classification separates the methods into broad classes based on the main optimization performed: adaptive resolution, geometric simplification, shader simplification, and chromatic degradation, as well spatio-temporal deterioration techniques provides. We've seen commonalities and differences among these methods, as well as specializations to specific setups, in particular concerning dynamic or static gaze points and raycasting and rasterization-based solutions. While the classes were well separated, we have also seen that it is not uncommon that actual solutions borrow methods from all of them, combining, e.g., the peripheral pixel undersampling of the adaptive resolution, with adaptive LODs for geometry, and spatio-temporal filters and caches.

The survey has also highlighted the substantial successes of these techniques, and their proven capability to drive a variety of applications. In terms of setups, moreover, while it was mostly applied to VR displays for a long time, recent years have seen an expansion towards near-eye AR and large high-resolution displays. With the current trend towards high-resolution displays covering large FOVs, it is expected that the technique will become more and more important.

However, despite the very significant successes and the potentially enormous gains of the method, it is still true that "foveated rendering is the holy grail in the modern computer graphics world, exciting but virtually elusive" [63]. This is mostly because, in order to really unleash its potential, foveation has to be applied very aggressively, which is extremely difficult, especially on large and complex scenes with photorealistic lighting. Moderate peripheral degradation has been shown to produce very high-quality experiences but also provides moderate advantages with respect to other non-foveated adaptive rendering techniques. Foveation gains start to be very effective when the central region is small and peripheral degradation is high. This is, however, not generally achievable without artifacts given today's state-of-the-art, as discussed in Sec. 10. We expect developments in both the computational and hardware-based solutions to eclipse today's best techniques in the near future, raising the standard of foveated rendered graphics to new heights.

Acknowledgments

This work was supported by the European Union's Horizon 2020 research and innovation programme "EVOCATION" under the Marie Skłodowska-Curie grant agreement No 813170. We thank the anonymous reviewers for their valuable comments and suggestions.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, *Youtube-8m: A large-scale video classification benchmark*, 2016.

[2] V. K. Adhikarla, A. Tariqul Islam, P. T. Kovács, and O. Staadt, *Fast and efficient data reduction approach for multi-camera light field display telepresence systems*, in 2013 3DTV Vision Beyond Depth (3DTV-CON), 2013, pp. 1–4.

[3] B. A. Ahlborn, O. Kreylos, B. Hamann, and O. Staadt, *A foveal inset for large display environments*, in IEEE Virtual Reality Conference (VR 2006), 2006, pp. 281–282.

[4] E. Akbas and M. P. Eckstein, *Object detection through search with a foveated visual system*, PLOS Computational Biology, 13 (2017), pp. 1–28.

[5] K. Akeley, *Reality engine graphics*, in Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '93, New York, NY, USA, 1993, Association for Computing Machinery, p. 109–116.

[6] T. Akenine-Möller, E. Haines, N. Hoffman, A. Pesce, M. Iwanicki, and S. Hillaire, *Real-Time Rendering 4th Edition*, A K Peters/CRC Press, Boca Raton, FL, USA, 2018.

[7] K. Akşit, P. Chakravarthula, K. Rathinavel, Y. Jeong, R. Albert, H. Fuchs, and D. Luebke, *Manufacturing application-driven foveated near-eye displays*, IEEE transactions on visualization and computer graphics, 25 (2019), pp. 1928–1939.

[8] R. Albert, A. Patney, D. Luebke, and J. Kim, *Latency requirements for foveated rendering in virtual reality*, ACM Trans. Appl. Percept., 14 (2017).

[9] R. A. Albert, A. Godinez, and D. Luebke, *Reading speed decreases for fast readers under gaze-contingent rendering*, in ACM Symposium on Applied Perception 2019, SAP '19, New York, NY, USA, 2019, Association for Computing Machinery.

[10] T. Ananpiriyakul, J. Anghel, K. Potter, and A. Joshi, *A gaze-contingent system for foveated multiresolution visualization of vector and volumetric data*, electronic imaging, 2020 (2020), pp. 374–1–374–11.

[11] S. R. Andersen, *The history of the ophthalmological society of copenhagen 1900–50*, Acta Ophthalmologica Scandinavica, 80 (2002), pp. 6–17.

[12] P. Andersson, J. Nilsson, T. Akenine-Möller, M. Oskarsson, K. Åström, and M. D. Fairchild, *Flip: A difference evaluator for alternating images*, Proc. ACM Comput. Graph. Interact. Tech., 3 (2020).

[13] C. Andrews, A. Endert, B. Yost, and C. North, *Information visualization on large, high-resolution displays: Issues, challenges, and opportunities*, Information Visualization, 10 (2011), p. 341–355.

[14] E. Arabadzhiyska, O. T. Tursun, K. Myszkowski, H.-P. Seidel, and P. Didyk, *Saccade landing position prediction for gaze-contingent rendering*, ACM Trans. Graph., 36 (2017).

[15] H. Aubert and R. Foerster, *Untersuchungen über den Raumsinn der Retina. II.: H.Aubert: Ueber die Grenzen der Farbenwahrnehmung auf dem seitlichen Theilen der Retina.*, 1857.

[16] T. Axblad, *Impact of foveated rendering on path tracing frame rate in head- mounted vr displays*, Master's thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2020.

[17] M. Ayama, M. Sakurai, O. Carlander, G. Derefeldt, and L. Eriksson, *Color appearance in peripheral vision*, in Human Vision and Electronic Imaging IX, B. E. Rogowitz and T. N. Pappas, eds., vol. 5292, International Society for Optics and Photonics, SPIE, 2004, pp. 260 – 271.

[18] J. Bae, J. Lee, and H. Nam, *Variable clock and em signal generation scheme for foveation-based driving oled head-mounted displays*, Electronics, 10 (2021).

[19] M. Balsa Rodriguez, E. Gobbetti, J. Iglesias Guitián, M. Makhinya, F. Marton, R. Pajarola, and S. Suter, *State-of-the-art in compressed gpu-based direct volume rendering*, Computer Graphics Forum, 33 (2014), pp. 77–100.

[20] G. R. Barnes, *Vestibulo-ocular function during co-ordinated head and eye movements to acquire visual targets.*, The Journal of Physiology, 287 (1979), pp. 127–147.

[21] B. A. Barsky, *Vision-realistic rendering: Simulation of the scanned foveal image from wavefront data of human subjects*, in Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization, APGV '04, New York, NY, USA, 2004, Association for Computing Machinery, p. 73–81.

[22] B. Bastani, E. Turner, C. Vieri, H. Jiang, B. Funt, and N. Balram, *Foveated pipeline for ar/vr head-mounted displays*, Information Display, 33 (2017), pp. 14–19 and 35.

[23] P. Baudisch, D. DeCarlo, A. T. Duchowski, and W. S. Geisler, *Focusing on the essential: Considering attention in display design*, Commun. ACM, 46 (2003), p. 60–66.

[24] P. Baudisch, N. Good, V. Bellotti, and P. Schraedley, *Keeping things in context: A comparative evaluation of focus plus context screens, overviews, and zooming*, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '02, New York, NY, USA, 2002, Association for Computing Machinery, p. 259–266.

[25] B. Behnam and T. Eric, *Google AI Blog introducing a new foveation pipeline for virtual/mixed reality*. `https://ai.googleblog.com/2017/12/introducing-new-foveation-pipeline-for.html`. Accessed: 2021-04-03.

[26] K. Bektaş, A. Çöltekin, J. Krüger, A. T. Duchowski, and S. I. Fabrikant, *Geogcd: Improved visual search via gaze-contingent display*, in Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA '19, New York, NY, USA, 2019, Association for Computing Machinery.

[27] K. Bektas, A. Cöltekin, J. Krüger, and A. T. Duchowski, *A testbed combining visual perception models for geographic gaze contingent displays*, in Eurographics Conference on Visualization (EuroVis) - Short Papers, E. Bertini, J. Kennedy, and E. Puppo, eds., The Eurographics Association, 2015.

[28] H. Benko, E. Ofek, F. Zheng, and A. D. Wilson, *Fovear: Combining an optically see-through near-eye display with projector-based spatial augmented reality*, in Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology, UIST '15, New York, NY, USA, 2015, Association for Computing Machinery, p. 129–135.

[29] S. Besenthal, S. Maisch, and T. Ropinski, *Multi-resolution rendering for computationally expensive lighting effects*, CoRR, abs/1906.04576 (2019).

[30] E. W. Bethel, B. Tierney, J. Lee, D. Gunter, and S. Lau, *Using high-speed wans and network data caches to enable remote and distributed visualization*, CoRR, abs/1801.09504 (2018).

[31] F. Bettio, E. Gobbetti, F. Marton, and G. Pintore, *Scalable rendering of massive triangle meshes on light field displays*, Computers & Graphics, 32 (2008), pp. 55–64.

[32] J. Beyer, M. Hadwiger, and H. Pfister, *State-of-the-art in gpu-based large-scale volume visualization*, Comput. Graph. Forum, 34 (2015), p. 13–37.

[33] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl, *Visualization of eye tracking data: A taxonomy and survey*, Computer Graphics Forum, 36 (2017), pp. 260–284.

[34] M. Böhme, M. Dorr, T. Martinetz, and E. Barth, *Gaze-contingent temporal filtering of video*, in Proceedings of the 2006 Symposium on Eye Tracking Research & Applications, ETRA '06, New York, NY, USA, 2006, Association for Computing Machinery, p. 109–115.

[35] N. D. B. Bruce and J. K. Tsotsos, *Saliency based on information maximization*, in Proceedings of the 18th International Conference on Neural Information Processing Systems, NIPS'05, Cambridge, MA, USA, 2005, MIT Press, p. 155–162.

[36] V. Bruder, C. Schulz, R. Bauer, S. Frey, D. Weiskopf, and T. Ertl, *Voronoi-based foveated volume rendering*, in EuroVis 2019 - Short Papers, J. Johansson, F. Sadlo, and G. E. Marai, eds., The Eurographics Association, 2019.

[37] S. L. Buck, R. Knight, G. Fowler, and B. Hunt, *Rod influence on hue-scaling functions*, Vision Research, 38 (1998), pp. 3259–3263.

[38] K. Cater, A. Chalmers, and P. Ledda, *Selective quality rendering by exploiting human inattentional blindness: Looking but not seeing*, in Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '02, New York, NY, USA, 2002, Association for Computing Machinery, p. 17–24.

[39] A. H. Chan and A. J. Courtney, *Foveal acuity, peripheral acuity and search performance: A review*, International Journal of Industrial Ergonomics, 18 (1996), pp. 113–119.

[40] D. Chandler, *Seven challenges in image quality assessment: Past, present, and future research*, International Scholarly Research Notices, 2013 (2013), pp. 1–53.

[41] C. Chang, W. Cui, and L. Gao, *Foveated holographic near-eye 3d display*, Opt. Express, 28 (2020), pp. 1345–1356.

[42] E.-C. Chang, S. Mallat, and C. Yap, *Wavelet foveation*, Applied and Computational Harmonic Analysis, 9 (2000), pp. 312–335.

[43] J. Chen, L. Mi, C. P. Chen, H. Liu, J. Jiang, and W. Zhang, *Design of foveated contact lens display for augmented reality*, Opt. Express, 27 (2019), pp. 38204–38219.

[44] I. Cheng, *Foveated 3d model simplification*, in Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings., vol. 1, 2003, pp. 241–244 vol.1.

[45] K. B. Chenliang Chang, W. Gordon, L. Byoungho, and L. Gao, *Toward the next-generation vr/ar optics: a review of holographic near-eye displays from a human-centric perspective*, Optica, 7 (2020), pp. 1563–1578.

[46] J. Choi and J. Ko, *Remotegl - towards low-latency interactive cloud graphics experience for mobile devices (demo)*, in Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '19, New York, NY, USA, 2019, Association for Computing Machinery, p. 693–694.

[47] P. Cignoni, F. Ganovelli, E. Gobbetti, F. Marton, F. Ponchio, and R. Scopigno, *Adaptive TetraPuzzles – efficient out-of-core construction and visualization of gigantic polygonal models*, ACM Transactions on Graphics, 23 (2004), pp. 796–803.

[48] A. Çöltekin and H. Haggrén, *Stereo foveation*, The Photogrammetric Journal of Finland, 20 (2006), pp. 45–54.

[49] R. L. Cook, T. Porter, and L. Carpenter, *Distributed ray tracing*, in Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '84, New York, NY, USA, 1984, Association for Computing Machinery, p. 137–145.

[50] E. Cuervo, K. Chintalapudi, and M. Kotaru, *Creating the perfect illusion: What will it take to create life-like virtual reality headsets?*, in Proceedings of the 19th International Workshop on Mobile Computing Systems & Applications, Hot-

Mobile '18, New York, NY, USA, 2018, Association for Computing Machinery, p. 7–12.

[51] E. Cuervo and D. Chu, *Poster: Mobile virtual reality for head-mounted displays with interactive streaming video and likelihood-based foveation*, in Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion, MobiSys '16 Companion, New York, NY, USA, 2016, Association for Computing Machinery, p. 130.

[52] R. J. Deeley, N. Drasdo, and W. N. Charman, *A simple parametric model of the human ocular modulation transfer function*, Ophthalmic and Physiological Optics, 11 (1991), pp. 91–93.

[53] H. Deubel, W. X. Schneider, et al., *Saccade target selection and object recognition: Evidence for a common attentional mechanism*, Vision research, 36 (1996), pp. 1827–1838.

[54] O. Deussen, M. Spicker, and Q. Zheng, *Weighted linde-buzo-gray stippling*, ACM Trans. Graph., 36 (2017).

[55] A. Deza, A. Jonnalagadda, and M. P. Eckstein, *Towards metamerism via foveated style transfer*, CoRR, abs/1705.10041 (2017).

[56] M. Dorr, T. Martinetz, M. Böhme, and E. Barth, *Visibility of temporal blur on a gaze-contingent display*, in Proceedings of the 2nd Symposium on Applied Perception in Graphics and Visualization, APGV '05, New York, NY, USA, 2005, Association for Computing Machinery, p. 33–36.

[57] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*, Springer-Verlag London, 01 2007.

[58] A. T. Duchowski, *Gaze-based interaction: A 30 year retrospective*, Computers & Graphics, 73 (2018), pp. 59 – 69.

[59] A. T. Duchowski, D. Bate, P. Stringfellow, K. Thakur, B. J. Melloy, and A. K. Gramopadhye, *On spatiochromatic visual sensitivity and peripheral color lod management*, ACM Trans. Appl. Percept., 6 (2009).

[60] A. T. Duchowski and A. Çöltekin, *Foveated gaze-contingent displays for peripheral lod management, 3d visualization, and stereo imaging*, ACM Trans. Multimedia Comput. Commun. Appl., 3 (2007).

[61] A. T. Duchowski, N. Cournia, and H. Murphy, *Gaze-contingent displays: A review*, CyberPsychology & Behavior, 7 (2004), pp. 621–634. PMID: 15687796.

[62] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[63] M. Finch, B. Guenter, and J. Snyder, *Foveated 3d display*, in ACM SIGGRAPH 2013 Emerging Technologies, SIGGRAPH '13, New York, NY, USA, 2013, Association for Computing Machinery.

[64] L. H. Frank, J. G. Casali, and W. W. Wierwille, *Effects of visual display and motion system delays on operator performance and uneasiness in a driving simulator*, Human Factors, 30 (1988), pp. 201–217. PMID: 3384446.

[65] L. Franke, L. Fink, J. Martschinke, K. Selgrad, and M. Stamminger, *Time-warped foveated rendering for virtual reality headsets*, Computer Graphics Forum, 40 (2021), pp. 110–123.

[66] L. Fridman, B. Jenik, S. Keshvari, B. Reimer, C. Zetzsche, and R. Rosenholtz, *Sideeye: A generative neural network based simulator of human peripheral vision*, 2017.

[67] F. Friess, M. Braun, V. Bruder, S. Frey, G. Reina, and T. Ertl, *Foveated encoding for large high-resolution displays*, IEEE Transactions on Visualization and Computer Graphics, 27 (2021), pp. 1850–1859.

[68] S. Friston, T. Ritschel, and A. Steed, *Perceptual rasterization for head-mounted display image synthesis*, ACM Trans. Graph., 38 (2019).

[69] M. Fujita and T. Harada, *Foveated real-time ray tracing for virtual reality headset*, Light Transport Entertainment Research, (2014).

[70] T. A. Funkhouser and C. H. Séquin, *Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments*, in Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '93, New York, NY, USA, 1993, Association for Computing Machinery, p. 247–254.

[71] C. Gao, Y. Peng, H. Li, and X. Liu, *Toward low-computation light field displays by foveated rendering*, in Optical Architectures for Displays and Sensing in Augmented, Virtual, and Mixed Reality (AR, VR, MR) II, B. C. Kress and C. Peroz, eds., vol. 11765, International Society for Optics and Photonics, SPIE, 2021, pp. 254 – 259.

[72] W. S. Geisler and J. S. Perry, *Real-time foveated multiresolution system for low-bandwidth video communication*, in Human Vision and Electronic Imaging III, B. E. Rogowitz and T. N. Pappas, eds., vol. 3299, International Society for Optics and Photonics, SPIE, 1998, pp. 294 – 305.

[73] W. S. Geisler and J. S. Perry, *Real-time simulation of arbitrary visual fields*, in Proceedings of the 2002 Symposium on Eye Tracking Research & Applications, ETRA '02, New York, NY, USA, 2002, Association for Computing Machinery, p. 83–87.

[74] A. S. Glassner, *Principles of digital image synthesis*, Elsevier, 2014.

[75] E. Gobbetti and F. Marton, *Layered point clouds – a simple and efficient multiresolution structure for distributing and rendering gigantic point-sampled models*, Computers & Graphics, 28 (2004), pp. 815–826.

[76] E. Gobbetti and F. Marton, *Far voxels: A multiresolution framework for interactive rendering of huge complex 3d models on commodity graphics platforms*, ACM Trans. Graph., 24 (2005), p. 878–885.

[77] G. Godin, J. françois Lalonde, and L. Borgeat, *Dual-resolution stereoscopic display with scene-adaptive fovea boundaries*, in in 8th International Immersive Projection Technology Workshop (to appear, 2004, pp. 13–14.

[78] G. Godin, P. Massicotte, and L. Borgeat, *High-resolution insets in projector-based stereoscopic displays: principles and techniques*, in Stereoscopic Displays and Virtual Reality Systems XIII, A. J. Woods, N. A. Dodgson, J. O. Merritt, M. T. Bolas, and I. E. McDowall, eds., vol. 6055, International Society for Optics and Photonics, SPIE, 2006, pp. 136 – 147.

[79] B. Guenter, *Motion compensated noise reduction*, Tech. Rep. MSR-TR-94-05, Microsoft Research, March 1994.

[80] B. Guenter, M. Finch, S. Drucker, D. Tan, and J. Snyder, *Foveated 3D graphics*, ACM Transactions on Graphics, 31 (2012).

[81] T. Hansen, L. Pracejus, and K. Gegenfurtner, *Color perception in the intermediate periphery of the visual field.*, Journal of vision, 9 4 (2009), pp. 26.1–12.

[82] Y. He, Y. Gu, and K. Fatahalian, *Extending the graphics pipeline with adaptive, multi-rate shading*, ACM Trans. Graph., 33 (2014).

[83] D. Hoffman, Z. Meraz, and E. Turner, *Limits of peripheral acuity and implications for vr system design*, Journal of the Society for Information Display, 26 (2018), pp. 483–495.

[84] J. Hong, *Foveation in near-eye holographic display*, in 2018 International Conference on Information and Communication Technology Convergence (ICTC), 2018, pp. 602–604.

[85] J. Hong, Y. Kim, S. Hong, C. Shin, and H. Kang, *Gaze contingent hologram synthesis for holographic head-mounted display*, in Practical Holography XXX: Materials and Applications, H. I. Bjelkhagen and V. M. B. Jr., eds., vol. 9771, International Society for Optics and Photonics, SPIE, 2016, pp. 117 – 122.

[86] D. Hood, *Lower-level visual processing and models of light adaptation*, Annual review of psychology, 49 (1998), pp. 503–535.

[87] E. Horvitz and J. Lengyel, *Perception, attention, and resources: A decision-theoretic approach to graphics rendering*, in Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, UAI'97, San Francisco, CA, USA, 1997, Morgan Kaufmann Publishers Inc., p. 238–249.

[88] C.-F. Hsu, A. Chen, C.-H. Hsu, C.-Y. Huang, C.-L. Lei, and K.-T. Chen, *Is foveated rendering perceivable in virtual*

reality? exploring the efficiency and consistency of quality assessment methods, in Proceedings of the 25th ACM International Conference on Multimedia, MM '17, New York, NY, USA, 2017, Association for Computing Machinery, p. 55–63.

[89] H. Hua, *Enabling focus cues in head-mounted displays*, Proceedings of the IEEE, 105 (2017), pp. 805–824.

[90] H. Hua and S. Liu, *Dual-sensor foveated imaging system*, Appl. Opt., 47 (2008), pp. 317–327.

[91] R. Hussain, M. Chessa, and F. Solari, *Mitigating cybersickness in virtual reality systems through foveated depth-of-field blur*, Sensors, 21 (2021).

[92] G. Illahi, M. Siekkinen, and E. Masala, *Foveated video streaming for cloud gaming*, in 2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP), 2017, pp. 1–6.

[93] G. K. Illahi, T. V. Gemert, M. Siekkinen, E. Masala, A. Oulasvirta, and A. Ylä-Jääski, *Cloud gaming with foveated graphics*, CoRR, abs/1809.05823 (2018).

[94] G. K. Illahi, T. V. Gemert, M. Siekkinen, E. Masala, A. Oulasvirta, and A. Ylä-Jääski, *Cloud gaming with foveated video encoding*, ACM Trans. Multimedia Comput. Commun. Appl., 16 (2020).

[95] H. J. Jang, J. Y. Lee, J. Kim, J. Kwak, and J.-H. Park, *Progress of display performances: Ar, vr, qled, and oled*, Journal of Information Display, 21 (2020), pp. 1–9.

[96] Q. Ji and X. Yang, *Real time visual cues extraction for monitoring driver vigilance*, in Computer Vision Systems, B. Schiele and G. Sagerer, eds., Berlin, Heidelberg, 2001, Springer Berlin Heidelberg, pp. 107–124.

[97] H. Jiang, T. Ning, and B. Bastani, *Efficient peripheral flicker reduction for foveated rendering in mobile vr systems*, in 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2020, pp. 802–803.

[98] Y. Jin, M. Chen, T. G. Bell, Z. Wan, and A. Bovik, *Study of 2D foveated video quality in virtual reality*, in Applications of Digital Image Processing XLIII, A. G. Tescher and T. Ebrahimi, eds., vol. 11510, International Society for Optics and Photonics, SPIE, 2020, pp. 18 – 26.

[99] Y. Jin, M. Chen, T. Goodall, A. Patney, and A. C. Bovik, *Subjective and objective quality assessment of 2d and 3d foveated video compression in virtual reality*, IEEE Transactions on Image Processing, (2021), pp. 1–1.

[100] H. Jisoo, K. Youngmin, H. Sunghee, S. Choonsung, and K. Hoonjong, *Near-eye foveated holographic display*, in Imaging and Applied Optics 2018 (3D, AO, AIO, COSI, DH, IS, LACSEA, LS&C, MATH, pcAOP), Optical Society of America, 2018, p. 3M2G.4.

[101] P. JOHANSSON, *Perceptually modulatedlevel of detail in real time graphics*, 2013.

[102] Josef Spjut, Ben Boudaoud, Jonghyun Kim, Trey Greer, Rachel Albert, Michael Stengel, Kaan Akşit, David Luebke, *Towards Standardized Classification of Foveated Displays*, IEEE Transactions on Visualization and Computer Graphics, 26 (May, 2020), pp. 2126–2134.

[103] Y.-G. Ju and J.-H. Park, *Foveated computer-generated hologram and its progressive update using triangular mesh scene model for near-eye displays*, Opt. Express, 27 (2019), pp. 23725–23738.

[104] K. Kanov, R. Burns, C. Lalescu, and G. Eyink, *The johns hopkins turbulence databases: an open simulation laboratory for turbulence research*, Computing in Science & Engineering, 17 (2015), pp. 10–17.

[105] D. Kanter, *Graphics processing requirements for enabling immersive vr*, 10 2015.

[106] A. S. Kaplanyan, A. Sochenov, T. Leimkühler, M. Okunev, T. Goodall, and G. Rufo, *Deepfovea: Neural reconstruction for foveated rendering and video compression using learned statistics of natural videos*, ACM Trans. Graph., 38 (2019).

[107] A. Kar and P. Corcoran, *A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms*, IEEE Access, 5 (2017), pp. 16495–16519.

[108] D. Kelly, *Spatial frequency selectivity in the retina*, Vision Research, 15 (1975), pp. 665–672.

[109] H. Kim, J. Yang, J. Lee, S. Yoon, Y. Kim, M. Choi, J. Yang, E. Ryu, and W. Park, *Eye tracking-based 360 vr foveated/tiled video rendering*, in 2018 IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2018, pp. 1–1.

[110] J. Kim, Y. Jeong, M. Stengel, K. Akşit, R. Albert, B. Boudaoud, T. Greer, J. Kim, W. Lopes, Z. Majercik, P. Shirley, J. Spjut, M. McGuire, and D. Luebke, *Foveated ar: Dynamically-foveated augmented reality display*, ACM Trans. Graph., 38 (2019).

[111] J. Kim, Q. Sun, F. Huang, L. Wei, D. Luebke, and A. Kaufman, *Perceptual studies for foveated light field displays*, ArXiv, abs/1708.06034 (2017).

[112] J.-H. Kim, K.-W. On, W. Lim, J. Kim, J.-W. Ha, and B.-T. Zhang, *Hadamard product for low-rank bilinear pooling*, arXiv preprint arXiv:1610.04325, (2016).

[113] V. Kiran Adhikarla, F. Marton, T. Balogh, and E. Gobbetti, *Real-time adaptive content retargeting for live multi-view capture and light field display*, The Visual Computer, 31 (2015), pp. 1023–1032.

[114] R. Konrad, A. Angelopoulos, and G. Wetzstein, *Gaze-contingent ocular parallax rendering for virtual reality*, ACM Trans. Graph., 39 (2020).

[115] M. Koskela, K. Immonen, T. Viitanen, P. Jääskeläinen, J. Multanen, and J. Takala, *Foveated instant preview for progressive rendering*, in SIGGRAPH Asia 2017 Technical Briefs, SA '17, New York, NY, USA, 2017, Association for Computing Machinery.

[116] M. Koskela, K. Immonen, T. Viitanen, P. Jääskeläinen, J. Multanen, and J. Takala, *Instantaneous foveated preview for progressive monte carlo rendering*, Computational Visual Media, 4 (2018), pp. 267–276.

[117] M. Koskela, A. Lotvonen, M. Mäkitalo, P. Kivi, T. Viitanen, and P. Jääskeläinen, *Foveated Real-Time Path Tracing in Visual-Polar Space*, in Eurographics Symposium on Rendering - DL-only and Industry Track, T. Boubekeur and P. Sen, eds., The Eurographics Association, 2019.

[118] M. Koskela, T. Viitanen, P. Jääskeläinen, and J. Takala, *Foveated path tracing*, in Advances in Visual Computing, G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, and T. Isenberg, eds., Cham, 2016, Springer International Publishing, pp. 723–732.

[119] G. A. Koulieris, K. Akşit, M. Stengel, R. K. Mantiuk, K. Mania, and C. Richardt, *Near-eye display and tracking technologies for virtual and augmented reality*, Computer Graphics Forum, 38 (2019), pp. 493–519.

[120] B. Krajancich, P. Kellnhofer, and G. Wetzstein, *A perceptual model for eccentricity-dependent spatio-temporal flicker fusion and its applications to foveated graphics*, ArXiv, abs/2104.13514 (2021).

[121] V. Labhishetty, S. A. Cholewiak, and M. S. Banks, *Contributions of foveal and non-foveal retina to the human eye's focusing response*, Journal of Vision, 19 (2019), pp. 18–18.

[122] S. Laine and T. Karras, *Efficient sparse voxel octrees*, IEEE Transactions on Visualization and Computer Graphics, 17 (2010), pp. 1048–1059.

[123] S. Lee, J. Cho, B. Lee, Y. Jo, C. Jang, D. Kim, and B. Lee, *Foveated retinal optimization for see-through near-eye multi-layer displays*, IEEE Access, 6 (2018), pp. 2170–2180.

[124] S. Lee, M. Pattichis, and A. Bovik, *Foveated video quality assessment*, IEEE Transactions on Multimedia, 4 (2002), pp. 129–132.

[125] J. Lemley, A. Kar, and P. Corcoran, *Eye tracking in augmented spaces: A deep learning approach*, in 2018 IEEE Games, Entertainment, Media Conference (GEM), 2018, pp. 1–6.

[126] J. Lemley, A. Kar, A. Drimbarean, and P. Corcoran, *Convolutional neural network implementation for eye-gaze*

estimation on low-quality consumer imaging systems, IEEE Transactions on Consumer Electronics, 65 (2019), pp. 179–187.

[127] M. Levoy and R. Whitaker, *Gaze-directed volume rendering*, SIGGRAPH Comput. Graph., 24 (1990), p. 217–223.

[128] D. Li, R. Du, A. Babu, C. D. Brumar, and A. Varshney, *A Log-Rectilinear Transformation for Foveated 360-degree Video Streaming*, IEEE Transactions on Visualization and Computer Graphics, (2021), pp. 1–1.

[129] R. Li, E. Whitmire, M. Stengel, B. Boudaoud, J. Kautz, D. Luebke, S. Patel, and K. Akşit, *Optical gaze tracking with spatially-sparse single-pixel detectors*, in 2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), 2020, pp. 117–126.

[130] W. Lin and C.-C. Jay Kuo, *Perceptual visual quality metrics: A survey*, Journal of Visual Communication and Image Representation, 22 (2011), pp. 297–312.

[131] T. Lindeberg, *Concealing rendering simplifications using gazecontingent depth of field*, Master's thesis, KTH, School of Computer Science and Communication (CSC), 2016.

[132] S. Liu and H. Hua, *Spatialchromatic foveation for gaze contingent displays*, in Proceedings of the 2008 Symposium on Eye Tracking Research & Applications, ETRA '08, New York, NY, USA, 2008, Association for Computing Machinery, p. 139–142.

[133] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum, *Learning to detect a salient object*, in 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[134] L. C. Loschky and G. W. McConkie, *User performance with gaze contingent multiresolutional displays*, in Proceedings of the 2000 Symposium on Eye Tracking Research & Applications, ETRA '00, New York, NY, USA, 2000, Association for Computing Machinery, p. 97–103.

[135] L. C. Loschky and G. S. Wolverton, *How late can you update gaze-contingent multiresolutional displays without detection?*, ACM Trans. Multimedia Comput. Commun. Appl., 3 (2007).

[136] D. Luebke and B. Hallen, *Perceptually driven simplification for interactive rendering*, in Proceedings of the 12th Eurographics Conference on Rendering, EGWR'01, Goslar, DEU, 2001, Eurographics Association, p. 223–234.

[137] D. Luebke, B. Hallen, D. Newfield, and B. Watson, *Perceptually driven simplification using gaze-directed rendering*, tech. rep., University of Virginia, 09 2000.

[138] P. Lungaro and K. Tollmar, *Eye-gaze based service provision and qoe optimization*, in Proc. 5th ISCA/DEGA Workshop on Perceptual Quality of Systems (PQS 2016), 2016, pp. 1–5.

[139] A. Maimone, A. Georgiou, and J. S. Kollin, *Holographic near-eye displays for virtual and augmented reality*, ACM Trans. Graph., 36 (2017).

[140] E. Malkin, A. Deza, and T. Poggio, *Cuda-optimized realtime rendering of a foveated visual system*, 2020.

[141] S. G. Mallat, *A theory for multiresolution signal decomposition: the wavelet representation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 11 (1989), pp. 674–693.

[142] R. Mantiuk, *Chapter 10 - gaze-dependent tone mapping for hdr video*, in High Dynamic Range Video, A. Chalmers, P. Campisi, P. Shirley, and I. G. Olaizola, eds., Academic Press, 2017, pp. 189–199.

[143] R. Mantiuk, *Gaze-dependent screen space ambient occlusion*, in Computer Vision and Graphics, L. J. Chmielewski, R. Kozera, A. Orłowski, K. Wojciechowski, A. M. Bruckstein, and N. Petkov, eds., Cham, 2018, Springer International Publishing, pp. 16–27.

[144] R. Mantiuk and S. Janus, *Gaze-dependent ambient occlusion*, in Advances in Visual Computing, G. Bebis, R. Boyle, B. Parvin, D. Koracin, C. Fowlkes, S. Wang, M.-H. Choi, S. Mantler, J. Schulze, D. Acevedo, K. Mueller, and M. Papka, eds., Berlin, Heidelberg, 2012, Springer Berlin Heidelberg, pp. 523–532.

[145] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, *Hdr-vdp-2: A calibrated visual metric for visibility and quality predictions in all luminance conditions*, ACM Trans. Graph., 30 (2011).

[146] R. Mantiuk and M. Markowski, *Gaze-dependent tone mapping*, in Image Analysis and Recognition, M. Kamel and A. Campilho, eds., Berlin, Heidelberg, 2013, Springer Berlin Heidelberg, pp. 426–433.

[147] R. K. Mantiuk, G. Denes, A. Chapiro, A. Kaplanyan, G. Rufo, R. Bachy, T. Lian, and A. Patney, *Fovvideovdp: A visible difference predictor for wide field-of-view video*, ACM Trans. Graph., 40 (2021).

[148] N.-X. Marianos, *Foveated rendering algorithms using eye-tracking technology in virtual reality*, diploma work, Technical University of Crete, 2018.

[149] R. Martin, *Specification and evaluation of level of detail selection criteria*, Virtual Real., 3 (1998), pp. 132–143.

[150] S. Matthews, A. Uribe-Quevedo, and A. Theodorou, *Rendering optimizations for virtual reality using eye-tracking*, in 2020 22nd Symposium on Virtual and Augmented Reality (SVR), 2020, pp. 398–405.

[151] M. Mauderer, S. Conte, M. A. Nacenta, and D. Vishwanath, *Depth perception with gaze-contingent depth of field*, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '14, New York, NY, USA, 2014, Association for Computing Machinery, p. 217–226.

[152] M. Mauderer, D. R. Flatla, and M. A. Nacenta, *Gaze-Contingent Manipulation of Color Perception*, Association for Computing Machinery, New York, NY, USA, 2016, p. 5191–5202.

[153] C. D. McMurrough, V. Metsis, J. Rich, and F. Makedon, *An eye tracking dataset for point of gaze detection*, in Proceedings of the Symposium on Eye Tracking Research and Applications, ETRA '12, New York, NY, USA, 2012, Association for Computing Machinery, p. 305–308.

[154] X. Meng, R. Du, J. F. JaJa, and A. Varshney, *3d-kernel foveated rendering for light fields*, IEEE Transactions on Visualization and Computer Graphics, (2020), pp. 1–1.

[155] X. Meng, R. Du, and A. Varshney, *Eye-dominance-guided foveated rendering*, IEEE Transactions on Visualization and Computer Graphics, 26 (2020), pp. 1972–1980.

[156] X. Meng, R. Du, M. Zwicker, and A. Varshney, *Kernel foveated rendering*, Proc. ACM Comput. Graph. Interact. Tech., 1 (2018).

[157] T. Mikami, K. Hirai, T. Nakaguchi, and N. Tsumura, *Realtime tone-mapping of high dynamic range image using gazing area information*, in Proc. International Conference on Computer and Information, 2010.

[158] I. S. Mohammadi, M. Hashemi, and M. Ghanbari, *An object-based framework for cloud gaming using player's visual attention*, in 2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW), 2015, pp. 1–6.

[159] R. A. A. Mohammed and O. Staadt, *Learning eye movements strategies on tiled large high-resolution displays using inverse reinforcement learning*, in 2015 International Joint Conference on Neural Networks (IJCNN), 2015, pp. 1–7.

[160] B. Mora, *Naive ray-tracing: A divide-and-conquer approach*, ACM Trans. Graph., 30 (2011).

[161] K. T. Mullen, *The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings.*, The Journal of physiology, 359 (1985), pp. 381–400.

[162] H. Murphy and A. Duchowski, *Gaze-contingent level of detail rendering*, EuroGraphics, 2001 (2001).

[163] H. A. Murphy, A. T. Duchowski, and R. A. Tyrrell, *Hybrid image/model-based gaze-contingent rendering*, ACM Trans. Appl. Percept., 5 (2009).

[164] K. Myszkowski, *Perception-based global illumination, rendering, and animation techniques*, in Proceedings of the 18th Spring Conference on Computer Graphics, SCCG '02, New York, NY, USA, 2002, Association for Computing Machinery, p. 13–24.

[165] D. Nehab, P. V. Sander, and J. R. Isidoro, *The real-time reprojection cache*, in ACM SIGGRAPH 2006 Sketches, SIGGRAPH '06, New York, NY, USA, 2006, Association for Com-

puting Machinery, p. 185–es.

[166] C. Noorlander, J. J. Koenderink, R. J. Den Olden, and B. W. Edens, *Sensitivity to spatiotemporal colour contrast in the peripheral visual field*, Vision Research, 23 (1983), pp. 1–11.

[167] T. Ohshima, H. Yamamoto, and H. Tamura, *Gaze-directed adaptive rendering for interacting with virtual space*, in Proceedings of the IEEE 1996 Virtual Reality Annual International Symposium, 1996, pp. 103–110.

[168] C. Papadopoulos and A. E. Kaufman, *Acuity-driven gigapixel visualization*, IEEE Transactions on Visualization and Computer Graphics, 19 (2013), pp. 2886–2895.

[169] S. Park, Y. I. Kim, and H. Nam, *Foveation-based reduced resolution driving scheme for immersive virtual reality displays*, Opt. Express, 27 (2019), pp. 29594–29605.

[170] D. Parkhurst and E. Niebur, *A feasibility test for perceptually adaptive level of detail rendering on desktop systems*, in Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization, APGV '04, New York, NY, USA, 2004, Association for Computing Machinery, p. 49–56.

[171] D. J. Parkhurst and E. Niebur, *Variable-resolution displays: A theoretical, practical, and behavioral evaluation*, Human Factors, 44 (2002), pp. 611–629. PMID: 12691369.

[172] A. Patney, J. Kim, M. Salvi, A. Kaplanyan, C. Wyman, N. Benty, A. Lefohn, and D. Luebke, *Perceptually-based foveated virtual reality*, in ACM SIGGRAPH 2016 Emerging Technologies, SIGGRAPH '16, New York, NY, USA, 2016, Association for Computing Machinery.

[173] A. Patney, M. Salvi, J. Kim, A. Kaplanyan, C. Wyman, N. Benty, D. Luebke, and A. Lefohn, *Towards foveated rendering for gaze-tracked virtual reality*, ACM Trans. Graph., 35 (2016).

[174] J. S. Perry and W. S. Geisler, *Gaze-contingent real-time simulation of arbitrary visual fields*, in Human Vision and Electronic Imaging VII, B. E. Rogowitz and T. N. Pappas, eds., vol. 4662, International Society for Optics and Photonics, SPIE, 2002, pp. 57 – 69.

[175] A. Peuhkurinen and T. Mikkonen, *Real-time human eye resolution ray tracing in mixed reality*, in Proc. GRAPP, 2021, pp. 169–176.

[176] S. Piccand, R. Noumeir, and E. Paquette, *Efficient visualization of volume data sets with region of interest and wavelets*, in Medical Imaging 2005: Visualization, Image-Guided Procedures, and Display, R. L. G. Jr. and K. R. Cleary, eds., vol. 5744, International Society for Optics and Photonics, SPIE, 2005, pp. 462 – 470.

[177] D. Pohl, T. Bolkart, S. Nickels, and O. Grau, *Using astigmatism in wide angle hmds to improve rendering*, in 2015 IEEE Virtual Reality (VR), 2015, pp. 263–264.

[178] D. Pohl, G. S. Johnson, and T. Bolkart, *Improved prewarping for wide angle, head mounted displays*, in Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology, VRST '13, New York, NY, USA, 2013, Association for Computing Machinery, p. 259–262.

[179] D. Pohl, X. Zhang, and A. Bulling, *Combining eye tracking with optimizations for lens astigmatism in modern wide-angle hmds*, in 2016 IEEE Virtual Reality (VR), 2016, pp. 269–270.

[180] D. Pohl, X. Zhang, A. Bulling, and O. Grau, *Concept for using eye tracking in a head-mounted display to adapt rendering to the user's current visual field*, in Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, VRST '16, New York, NY, USA, 2016, Association for Computing Machinery, p. 323–324.

[181] D. Purves, G. J. Augustine, D. Fitzpatrick, L. C. Katz, A.-S. LaMantia, J. O. McNamara, S. M. Williams, et al., *Types of eye movements and their functions*, Neuroscience, (2001), pp. 361–390.

[182] R. Radkowski and S. Raul, *Impact of foveated rendering on procedural task training*, in Virtual, Augmented and Mixed Reality. Multimodal Interaction, J. Y. Chen and G. Fragomeni, eds., Cham, 2019, Springer International Publishing, pp. 258–267.

[183] M. Reddy, *Perceptually modulated level of detail for virtual environments*, PhD thesis, University of Edinburgh. College of Science and Engineering. School of ..., 1997.

[184] M. Reddy, *Perceptually optimized 3d graphics*, IEEE Computer Graphics and Applications, 21 (2001), pp. 68–75.

[185] S. M. Reder, *On-line monitoring of eye-position signals in contingent and noncontingent paradigms*, Behavior Research Methods & Instrumentation, 5 (1973), pp. 218–228.

[186] E. Reinhard, G. Ward, S. Pattanaik, and P. Debevec, *High Dynamic Range Imaging: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[187] R. A. Rensink, *Change detection*, Annual Review of Psychology, 53 (2002), pp. 245–277. PMID: 11752486.

[188] S. Rimac-Drlje, M. Vranjes, and D. Zagar, *Foveated mean squared error—a novel video quality metric*, Multimedia Tools and Applications, 49 (2009), pp. 425–445.

[189] M. F. Romero-Rondón, L. Sassatelli, F. Precioso, and R. Aparicio-Pardo, *Foveated streaming of virtual reality videos*, in Proceedings of the 9th ACM Multimedia Systems Conference, MMSys '18, New York, NY, USA, 2018, Association for Computing Machinery, p. 494–497.

[190] T. Roth, M. Weier, A. Hinkenjann, Y. Li, and P. Slusallek, *An analysis of eye-tracking data in foveated ray tracing*, in 2016 IEEE Second Workshop on Eye Tracking and Visualization (ETVIS), 2016, pp. 69–73.

[191] T. Roth, M. Weier, A. Hinkenjann, Y. Li, and P. Slusallek, *A quality-centered analysis of eye tracking data in foveated rendering*, Journal of Eye Movement Research (JEMR), 10 (2017).

[192] T. Roth, M. Weier, J. Maiero, A. Hinkenjann, and Y. Li, *Guided high-quality rendering*, in Advances in Visual Computing, G. Bebis, R. Boyle, B. Parvin, D. Koracin, I. Pavlidis, R. Feris, T. McGraw, M. Elendt, R. Kopper, E. Ragan, Z. Ye, and G. Weber, eds., Cham, 2015, Springer International Publishing, pp. 115–125.

[193] J. Ryoo, K. Yun, D. Samaras, S. R. Das, and G. Zelinsky, *Design and evaluation of a foveated video streaming service for commodity client devices*, in Proc. International Conference on Multimedia Systems, 2016, pp. 1–11.

[194] M. Sakurai, M. Ayama, and T. Kumagai, *Color appearance in the entire visual field: color zone map based on the unique hue component*, J. Opt. Soc. Am. A, 20 (2003), pp. 1997–2009.

[195] C. Scheel, F. Löffler, A. Lehmann, H. Schumann, and O. Staadt, *Dynamic level of detail for tiled large high-resolution displays*, in Proc. Virtuelle Und Erweiterte Realität 2014, Berichte Aus Der Informatik, Shaker Verlag, 2014, pp. 109–119.

[196] D. Scherzer, L. Yang, O. Mattausch, D. Nehab, P. V. Sander, M. Wimmer, and E. Eisemann, *A survey on temporal coherence methods in real-time rendering*, in EUROGRAPHICS 2011 State of the Art Reports, Eurographics Association, 2011, pp. 101–126.

[197] R. Shea, J. Liu, E. C.-H. Ngai, and Y. Cui, *Cloud gaming: architecture and performance*, IEEE network, 27 (2013), pp. 16–21.

[198] H. R. Sheikh, B. L. Evans, and A. C. Bovik, *Real-time foveation techniques for low bit rate video coding*, Real-Time Imaging, 9 (2003), pp. 27–40.

[199] S. Shimizu, *Wide-angle foveation for all-purpose use*, IEEE/ASME Transactions on Mechatronics, 13 (2008), pp. 587–597.

[200] A. Siekawa, M. Chwesiuk, R. Mantiuk, and R. Piórkowski, *Foveated ray tracing for vr headsets*, in MultiMedia Modeling, I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W.-H. Cheng, and S. Vrochidis, eds., Cham, 2019, Springer International Publishing, pp. 106–117.

[201] A. Siekawa and S. R. Mantiuk, *Gaze-dependent ray tracing*, in Proc. CESCG 2014: The 18th Central European Seminar on Computer Graphics, 2014.

[202] J. Spjut and B. Boudaoud, *Foveated displays: Toward classification of the emerging field*, in ACM SIGGRAPH 2019 Talks,

SIGGRAPH '19, New York, NY, USA, 2019, Association for Computing Machinery.
[203] M. Stengel, S. Grogorick, M. Eisemann, and M. Magnor, *Adaptive image-space sampling for gaze-contingent real-time rendering*, Computer Graphics Forum, 35 (2016), pp. 129–139.
[204] M. Stengel and M. Magnor, *Gaze-contingent computational displays: Boosting perceptual fidelity*, IEEE Signal Processing Magazine, 33 (2016), pp. 139–148.
[205] w. steve, *Variable-rate shading (VRS) - Win32 apps*, Apr. 2019.
[206] H. Strasburger, I. Rentschler, and M. Jüttner, *Peripheral vision and pattern recognition: A review*, Journal of vision, 11 (2011), p. 13.
[207] Q. Sun, F.-C. Huang, J. Kim, L.-Y. Wei, D. Luebke, and A. Kaufman, *Perceptually-guided foveation for light field displays*, ACM Trans. Graph., 36 (2017).
[208] I. E. Sutherland, R. F. Sproull, and R. A. Schumacker, *A characterization of ten hidden-surface algorithms*, ACM Computing Surveys, 6 (1974), pp. 1–55.
[209] N. T. Swafford, D. Cosker, and K. Mitchell, *Latency aware foveated rendering in unreal engine 4*, in Proceedings of the 12th European Conference on Visual Media Production, CVMP '15, New York, NY, USA, 2015, Association for Computing Machinery.
[210] N. T. Swafford, J. A. Iglesias-Guitian, C. Koniaris, B. Moon, D. Cosker, and K. Mitchell, *User, metric, and computational evaluation of foveated rendering methods*, in Proceedings of the ACM Symposium on Applied Perception, SAP '16, New York, NY, USA, 2016, Association for Computing Machinery, p. 7–14.
[211] M. F. Syawaludin, M. Lee, and J.-I. Hwang, *Foveation pipeline for 360° video-based telemedicine*, Sensors (Basel, Switzerland), 20 (2020).
[212] R. Tamstorf and H. Pritchett, *The challenges of releasing the moana island scene*, in Proc. EG Symposium on Rendering - Industrial track, 2019, pp. 73–74.
[213] G. Tan, Y.-H. Lee, T. Zhan, J. Yang, S. Liu, D. Zhao, and S.-T. Wu, *Foveated imaging for near-eye displays*, Opt. Express, 26 (2018), pp. 25076–25085.
[214] G. Tan, Y.-H. Lee, T. Zhan, J. Yang, S. Liu, D. Zhao, and S.-T. Wu, *45-4: Near-eye foveated display for achieving human visual acuity*, SID Symposium Digest of Technical Papers, 50 (2019), pp. 624–627.
[215] A. T.Duchowski, *A breadth-first survey of eye-tracking applications*, Behav Res Methods Instrum Comput., 34(4) (2002), pp. 455–70.
[216] L. N. Thibos, F. E. Cheney, and D. J. Walsh, *Retinal limits to the detection and resolution of gratings*, J. Opt. Soc. Am. A, 4 (1987), pp. 1524–1529.
[217] V. Thumuluri and M. Sharma, *A unified deep learning approach for foveated rendering novel view synthesis from sparse rgb-d light fields*, in 2020 International Conference on 3D Immersion (IC3D), 2020, pp. 1–8.
[218] R. Thunström, *Passive gaze-contingent techniques relation to system latency*, Master's thesis, Blekinge Institute of Technology, 2014.
[219] A. Tiwary, M. Ramanathan, and J. Kosinka, *Accelerated foveated rendering based on adaptive tessellation*, in Eurographics 2020 - Short Papers, A. Wilkie and F. Banterle, eds., The Eurographics Association, 2020.
[220] W. Tsai and Y. Liu, *Foveation-based image quality assessment*, in 2014 IEEE Visual Communications and Image Processing Conference, 2014, pp. 25–28.
[221] Y.-J. Tsai, Y.-X. Wang, and M. Ouhyoung, *Affordable system for measuring motion-to-photon latency of virtual reality in mobile devices*, in SIGGRAPH Asia 2017 Posters, SA '17, New York, NY, USA, 2017, Association for Computing Machinery.
[222] E. Turner, H. Jiang, D. Saint-Macary, and B. Bastani, *Phase-aligned foveated rendering for virtual reality headsets*, in 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), 2018, pp. 1–2.
[223] O. T. Tursun, E. Arabadzhiyska-Koleva, M. Wernikowski, R. Mantiuk, H.-P. Seidel, K. Myszkowski, and P. Didyk, *Luminance-contrast-aware foveated rendering*, ACM Trans. Graph., 38 (2019).
[224] C. W. Tyler, *Analysis of visual modulation sensitivity. iii. meridional variations in peripheral flicker sensitivity*, J. Opt. Soc. Am. A, 4 (1987), pp. 1612–1619.
[225] K. Vaidyanathan, M. Salvi, R. Toth, T. Foley, T. Akenine-Möller, J. Nilsson, J. Munkberg, J. Hasselgren, M. Sugihara, P. Clarberg, T. Janczak, and A. Lefohn, *Coarse Pixel Shading*, in Eurographics/ ACM SIGGRAPH Symposium on High Performance Graphics, I. Wald and J. Ragan-Kelley, eds., The Eurographics Association, Sep 2014.
[226] C. Vieri, G. Lee, N. Balram, S. H. Jung, J. Y. Yang, S. Y. Yoon, and I. B. Kang, *An 18 megapixel 4.3 inch 1443 ppi 120 hz oled display for wide field of view high acuity head mounted displays*, Journal of the Society for Information Display, 26 (2018), pp. 314–324.
[227] I. Viola, A. Kanitsar, and M. Groller, *Importance-driven volume rendering*, in IEEE Visualization 2004, 2004, pp. 139–145.
[228] M. Vranješ, S. Rimac-Drlje, and D. Vranješ, *Foveation-based content adaptive root mean squared error for video quality assessment*, Multimedia Tools and Applications, 77 (2018), pp. 21053–21082.
[229] Z. Wang and A. Bovik, *Foveated image and video coding*, Signal Processing and Communications, (2005).
[230] Z. Wang, A. C. Bovik, L. Lu, and J. L. Kouloheris, *Foveated wavelet image quality index*, in Applications of Digital Image Processing XXIV, A. G. Tescher, ed., vol. 4472, International Society for Optics and Photonics, SPIE, 2001, pp. 42 – 52.
[231] A. Watson, *A formula for human retinal ganglion cell receptive field density as a function of visual field location*, Journal of vision, 14 (2014).
[232] B. Watson, N. Walker, and L. Hodges, *A user study evaluating level of detail degradation in the periphery of head-mounted displays*, in Framework for Interactive Virtual Environments (FIVE) Conference, 03 1996.
[233] B. Watson, N. Walker, and L. F. Hodges, *Effectiveness of spatial level of detail degradation in the periphery of head-mounted displays*, in Conference Companion on Human Factors in Computing Systems, CHI '96, New York, NY, USA, 1996, Association for Computing Machinery, p. 227–228.
[234] B. Watson, N. Walker, and L. F. Hodges, *Supra-threshold control of peripheral LOD*, ACM Transactions on Graphics (TOG), 23 (2004), pp. 750–759.
[235] B. Watson, N. Walker, L. F. Hodges, and A. Worden, *Managing level of detail through peripheral degradation: Effects on search performance with a head-mounted display*, ACM Trans. Comput.-Hum. Interact., 4 (1997), p. 323–346.
[236] K. Weaver, *Design and evaluation of a perceptually adaptive rendering system for immersive virtual reality environments*, Master's thesis, Digital Repository @ Iowa State University, http://lib.dr.iastate.edu/, 2007.
[237] M. Webster, K. Halen, A. J. Meyers, P. Winkler, and J. Werner, *Colour appearance and compensation in the near periphery*, Proceedings of the Royal Society B: Biological Sciences, 277 (2010), pp. 1817 – 1825.
[238] L. Wei and Y. Sakamoto, *Fast calculation method with foveated rendering for computer-generated holograms using an angle-changeable ray-tracing method*, Appl. Opt., 58 (2019), pp. A258–A266.
[239] M. Weier, *Perception-driven rendering : techniques for the efficient visualization of 3D scenes including view- and gaze-contingent approaches*, PhD thesis, Saarland University, Saarbrücken, 2019.
[240] M. Weier, J. Maiero, T. Roth, A. Hinkenjann, and P. Slusallek, *Enhancing rendering performance with view-direction-based rendering techniques for large, high resolution multi-display systems*, in 11. Workshop Virtuelle Realität und

Augmented Reality der GI-Fachgruppe VR/AR, September 2014.

[241] M. Weier, J. Maiero, T. Roth, A. Hinkenjann, and P. Slusallek, *Lazy details for large high-resolution displays*, in SIGGRAPH Asia 2014 Posters, SA '14, New York, NY, USA, 2014, Association for Computing Machinery.

[242] M. Weier, T. Roth, A. Hinkenjann, and P. Slusallek, *Foveated depth-of-field filtering in head-mounted displays*, ACM Trans. Appl. Percept., 15 (2018).

[243] M. Weier, T. Roth, E. Kruijff, A. Hinkenjann, A. Pérard-Gayot, P. Slusallek, and Y. Li, *Foveated real-time ray tracing for head-mounted displays*, Comput. Graph. Forum, 35 (2016), p. 289–298.

[244] M. Weier, M. Stengel, T. Roth, P. Didyk, E. Eisemann, M. Eisemann, S. Grogorick, A. Hinkenjann, E. Kruijff, M. Magnor, K. Myszkowski, and P. Slusallek, *Perception-driven accelerated rendering*, Comput. Graph. Forum, 36 (2017), p. 611–643.

[245] F. W. Weymouth, *Visual sensory units and the minimal angle of resolution**, American Journal of Ophthalmology, 46 (1958), pp. 102–113.

[246] S. Winkler, M. Kunt, and C. J. van den Branden Lambrecht, *Vision and Video: Models and Applications*, Springer US, 2001, pp. 201–229.

[247] J.-Y. Wu and J. Kim, *Prescription ar: a fully-customized prescription-embedded augmented reality display*, Opt. Express, 28 (2020), pp. 6225–6241.

[248] K. Xiao, G. Liktor, and K. Vaidyanathan, *Coarse pixel shading with temporal supersampling*, in Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, I3D '18, New York, NY, USA, 2018, Association for Computing Machinery.

[249] L. Xiao, A. Kaplanyan, A. Fix, M. Chapman, and D. Lanman, *Deepfocus: Learned image synthesis for computational displays*, ACM Trans. Graph., 37 (2018).

[250] L. Xiao, S. Nouri, M. Chapman, A. Fix, D. Lanman, and A. Kaplanyan, *Neural supersampling for real-time rendering*, ACM Trans. Graph., 39 (2020).

[251] J. Xiong, G. Tan, T. Zhan, and S.-T. Wu, *Breaking the field-of-view limit in augmented reality with a scanning waveguide display*, OSA Continuum, 3 (2020), pp. 2730–2740.

[252] T. Yamauchi, T. Mikami, O. Ouda, T. Nakaguchi, and N. Tsumura, *Improvement and evaluation of real-time tone mapping for high dynamic range images using gaze information*, in Computer Vision – ACCV 2010 Workshops, R. Koch and F. Huang, eds., Berlin, Heidelberg, 2011, Springer Berlin Heidelberg, pp. 440–449.

[253] J. Yang, X. Li, and A. G. Campbell, *Variable rate ray tracing for virtual reality*, in SIGGRAPH Asia 2020 Posters, SA '20, New York, NY, USA, 2020, Association for Computing Machinery.

[254] L. Yang, D. Nehab, P. V. Sander, P. Sitthi-amorn, J. Lawrence, and H. Hoppe, *Amortized supersampling*, ACM Trans. Graph., 28 (2009), p. 1–12.

[255] Y. Ye, J. M. Boyce, and P. Hanhart, *Omnidirectional 360° Video Coding Technology in Responses to the Joint Call for Proposals on Video Compression With Capability Beyond HEVC*, IEEE Transactions on Circuits and Systems for Video Technology, 30 (2020), pp. 1241–1252.

[256] C. Yoo, J. Xiong, S. Moon, D. Yoo, C.-K. Lee, S.-T. Wu, and B. Lee, *Foveated display system based on a doublet geometric phase lens*, Opt. Express, 28 (2020), pp. 23690–23702.

[257] S.-e. Yoon, E. Gobbetti, D. Kasik, and D. Manocha, *Real-time massive model rendering*, Synthesis Lectures on Computer Graphics and Animation, 2 (2008), pp. 1–122.

[258] H. Yu, E. Chang, Z. Huang, and Z. Zheng, *Fast rendering of foveated volumes in wavelet-based representation*, Vis. Comput., 21 (2005), pp. 735–744.

[259] L. Zhang, R. Albert, J. Kim, and D. Luebke, *Developing a peripheral color tolerance model for gaze-contingent rendering*, Journal of Vision, 19 (2019), pp. 298c–298c.

[260] X. Zhang, W. Chen, Z. Yang, C. Zhu, and Q. Peng, *A new foveation ray casting approach for real-time rendering of 3d scenes*, in 2011 12th International Conference on Computer-Aided Design and Computer Graphics, 2011, pp. 99–102.

[261] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling, *Appearance-based gaze estimation in the wild*, in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015, pp. 4511–4520.

[262] Z. Zheng, Z. Yang, Y. Zhan, Y. Li, and W. Yu, *Perceptual model optimized efficient foveated rendering*, in Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology, VRST '18, New York, NY, USA, 2018, Association for Computing Machinery.

[263] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, *Learning deep features for scene recognition using places database*, in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds., vol. 27, Curran Associates, Inc., 2014.